

# Version Control with Git

Pete Alonzi

Research Data Services UVa Library

March 24, 2016

# Check us out data.library.virginia.edu

University of Virginia Library

## Research Data Services

**Research Data Services** offers consultation and training in acquiring, collecting, analyzing, visualizing, sharing, and preserving research data. Email us at [researchdataservices@virginia.edu](mailto:researchdataservices@virginia.edu).



[Spring Workshops](#)

[StatLab Articles](#)

### StatLab: Data Analysis & Statistics

- Consultation and training on data analysis, statistical methods, visualization, data wrangling, and the use of statistical software
- Contact: [statlab@virginia.edu](mailto:statlab@virginia.edu)

### Geospatial Analysis & Visualization

- Consultation, training and software support in spatial data and analysis, cartography and Esri products
- Contact: [uvagis@virginia.edu](mailto:uvagis@virginia.edu)

Research Software Support



# Installation Instructions

- Windows: <http://msysgit.github.io/>
- Mac: type git at terminal and click install
  - If that fails goto: <http://alonzi.github.io/2015-04-16-uvalib/>

# What we're gonna do today

- Version Control Software
  - principles
  - workflow
- Git repositories
  - Implementation of VCS principles
  - commands
- Collaboration (remote repositories)
  - Github
  - conflicts

# What version control does

- Record all versions of code
- Permit switching of versions
- Allow collaboration
- (Easy to use) ... if you build the habit

Good for text based files. Not so much for things like pdfs.

# "FINAL".doc



FINAL.doc!



FINAL\_rev.2.doc



FINAL\_rev.6.COMMENTS.doc

FINAL\_rev.8.comments5.  
CORRECTIONS.doc



FINAL\_rev.18.comments7.  
corrections9.MORE.30.doc



FINAL\_rev.22.comments49.  
corrections.10.#%\$%WHYDID  
ICOMETOGRADSCHOOL?????.doc

JORGE CHAM © 2012

# Solutions to this problem



**git**

**BITKEEPER**  
SOURCE MANAGEMENT



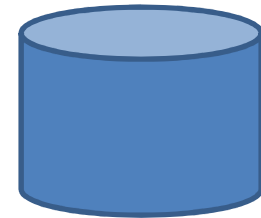
# Taxonomy

- Local data model (SCCS)
  - shared file system
- Client-server model (CVS)
  - singular reference file system
- Distributed model (Git)
  - All work is on local file system (not shared)

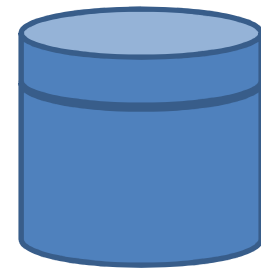


# Terminology

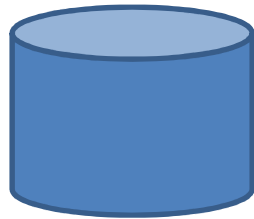
- “**Repository**:A directory where Git has been initialized to start version controlling your files.” - <https://try.github.io/levels/1/challenges/1>



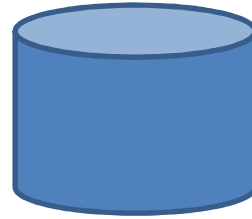
- **Revision**: An entry in the repository.



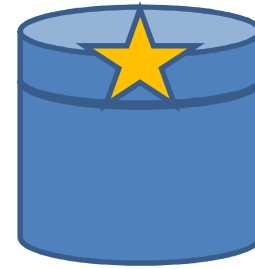
# Version Control Workflow



load



modify



record

# Git 101

- The way git works is by issuing commands at the command line.
- They look like this “git *verb*”

eg

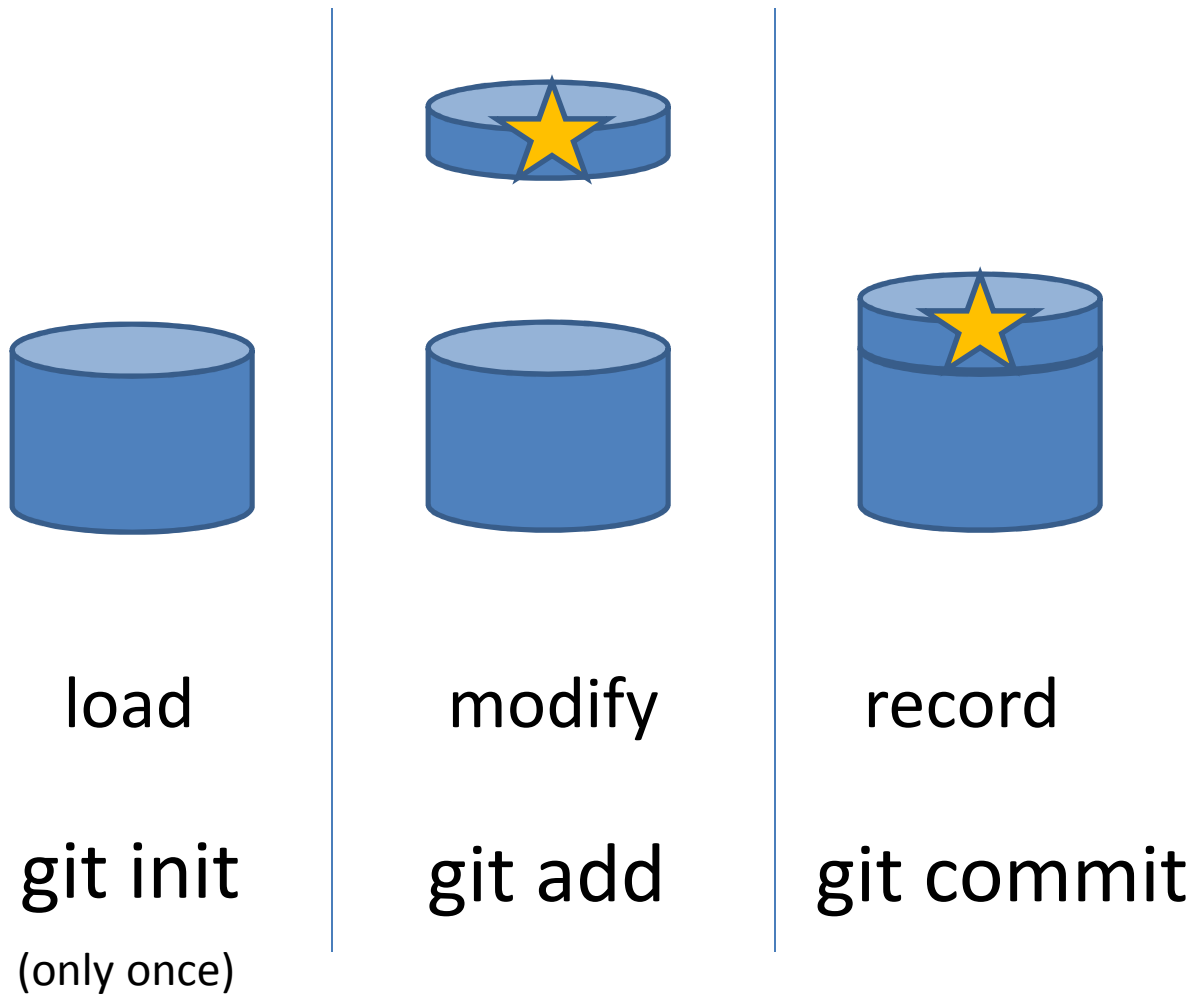
git commit

git add

git rm

git revert

# Git workflow



- Check status – `git status`

# Jedi Code

- *There is no emotion, there is peace.*
- *There is no ignorance, there is knowledge.*
- *There is no passion, there is serenity.*
- *There is no chaos, there is harmony.*
- *There is no death, there is the Force.*

Exercise 1

Create a repository

Add a text file

Make five commits

# More commands

- `git log`
- `git diff [revision]`

# Dangerous commands

- `git mv`
- `git rm`
  
- `git checkout [revision]`
- `git checkout master` (to get back)

# Sith Code

- *Peace is a lie, there is only passion.*
- *Through passion, I gain strength.*
- *Through strength, I gain power.*
- *Through power, I gain victory.*
- *Through victory, my chains are broken.*
- *The Force shall free me.*

Exercise 2

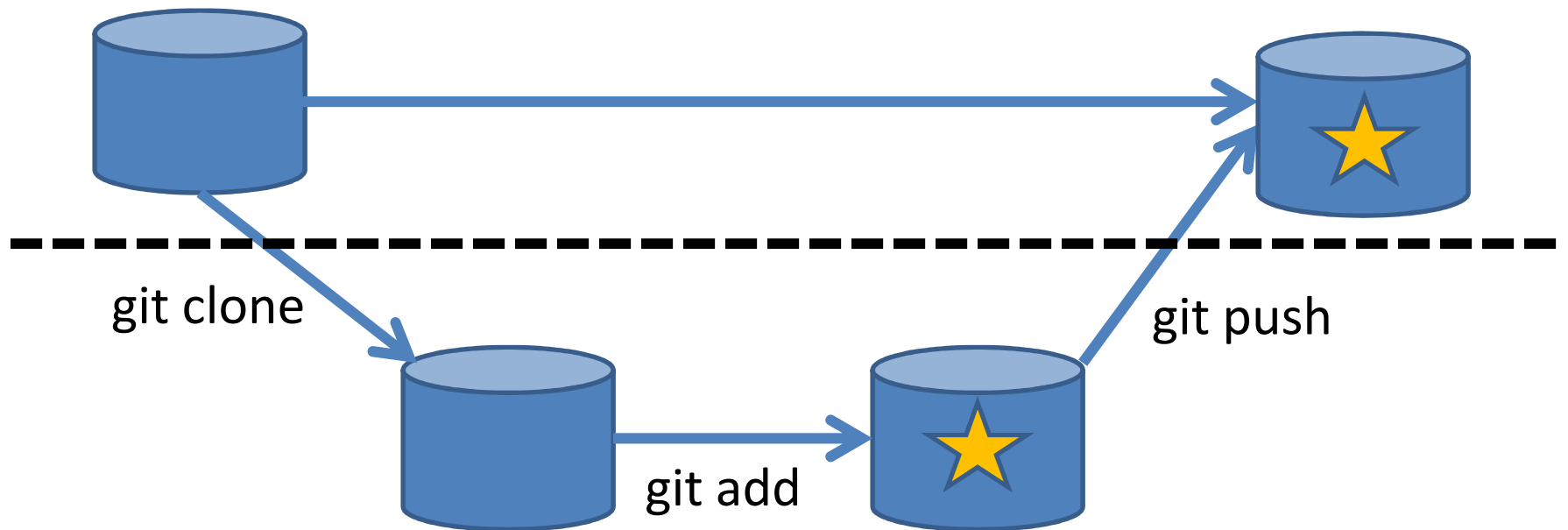
Add Sith Code to repository



# Remote repositories

# Remote repository

## Computer A



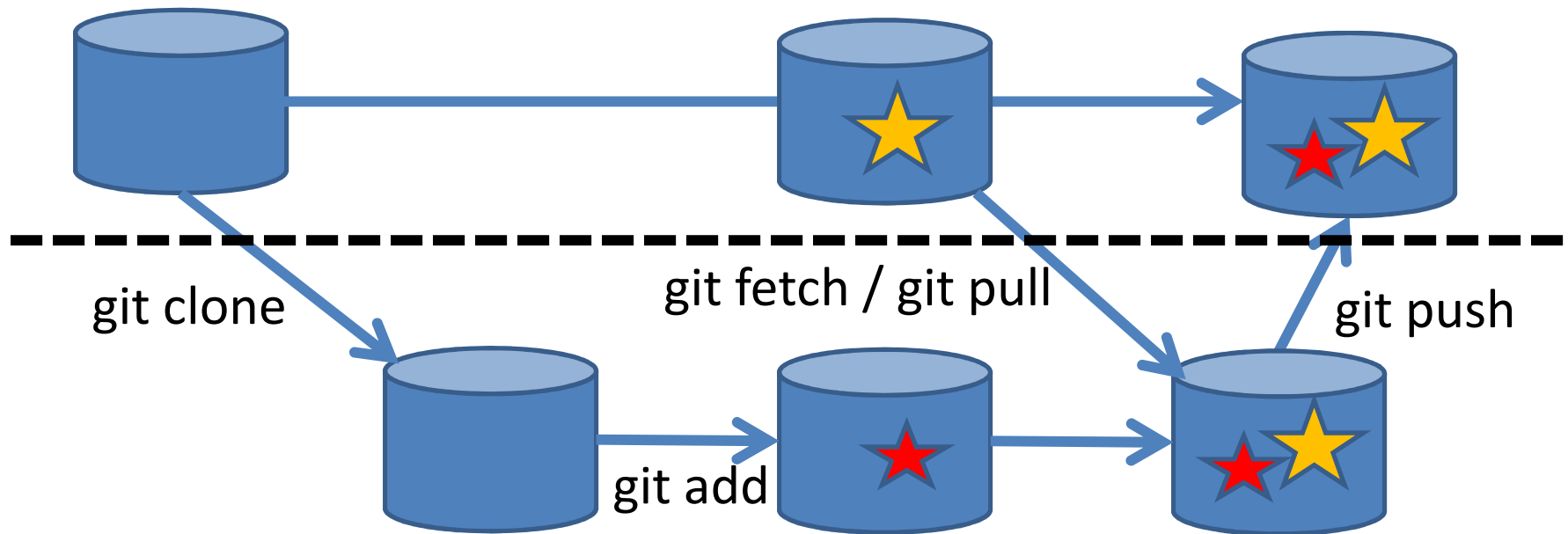
## Computer B

# Remote repository clone

- mkdir code
- cd code
- git clone <https://github.com/alonzi/gitdemo.git>

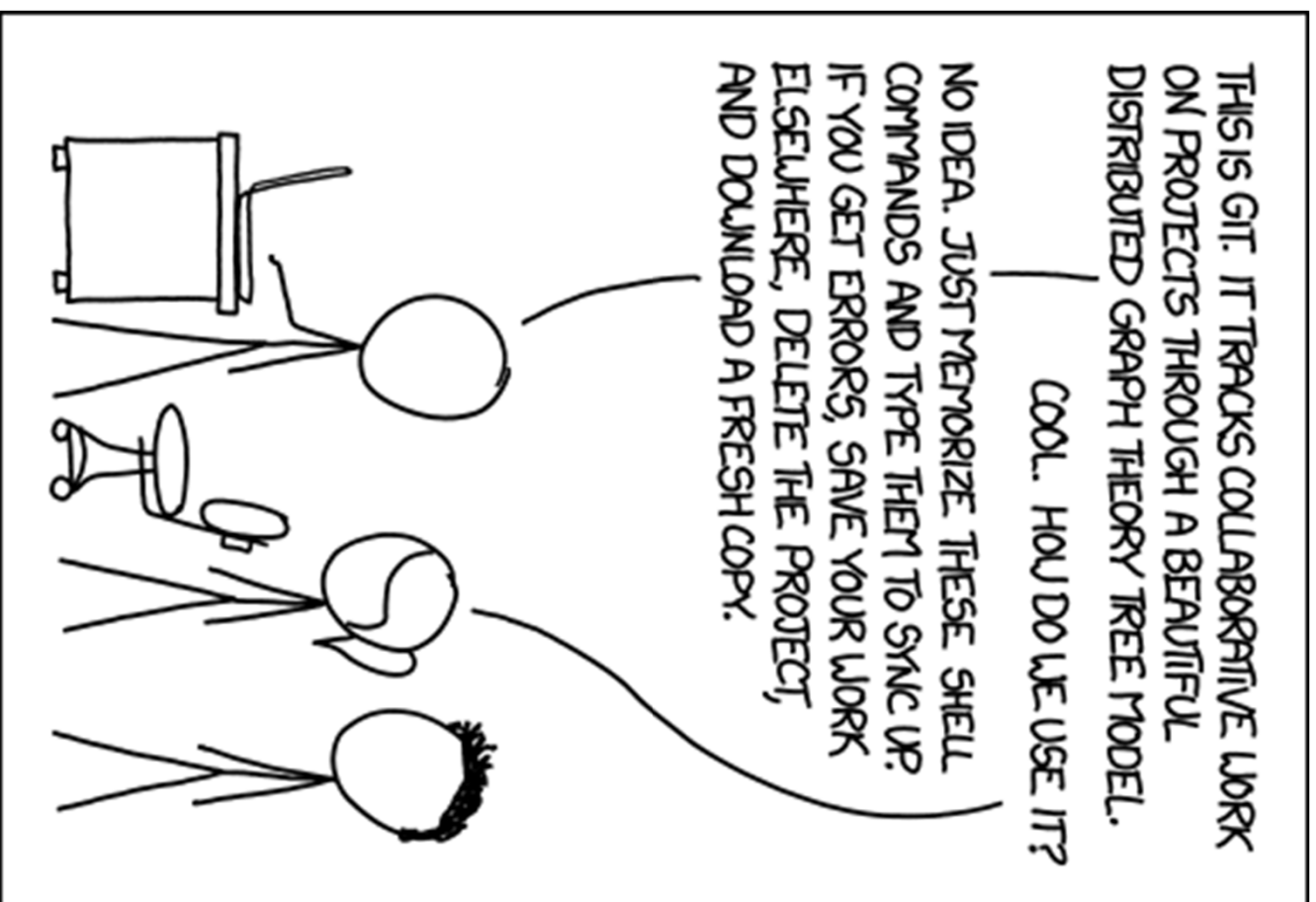
# Remote repository II

## Computer A



## Computer B

# CONCLUSIONS



extras

# git persona

- `git config --global user.name "Vlad Dracula"`
- `$ git config --global user.email "vlad@tran.sylvan.ia"`
- `$ git config --global color.ui "auto"`
- `$ git config --global core.editor "editor"`

From:

<http://swcarpentry.github.io/git-novice/01-setup.html>

# Ignoring file

- Create a file `.gitignore`
- Put command line style statements in it
- Eg: `*.txt` or `directory/`
  
- Git will now ignore the following files



- ned