

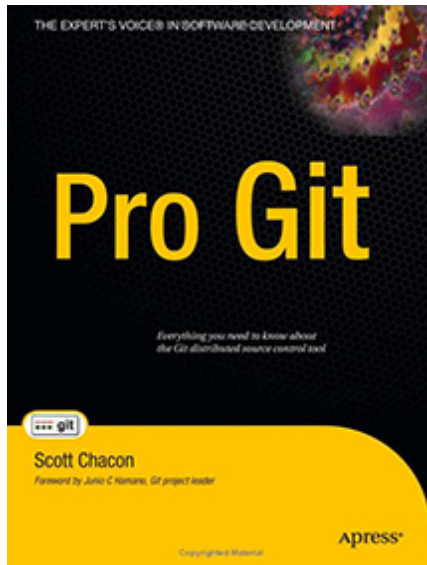
# Source Code Mgmt. Basics

**Harry J. Wang, Ph.D.**

University of Delaware

Fall 2017

# Resources



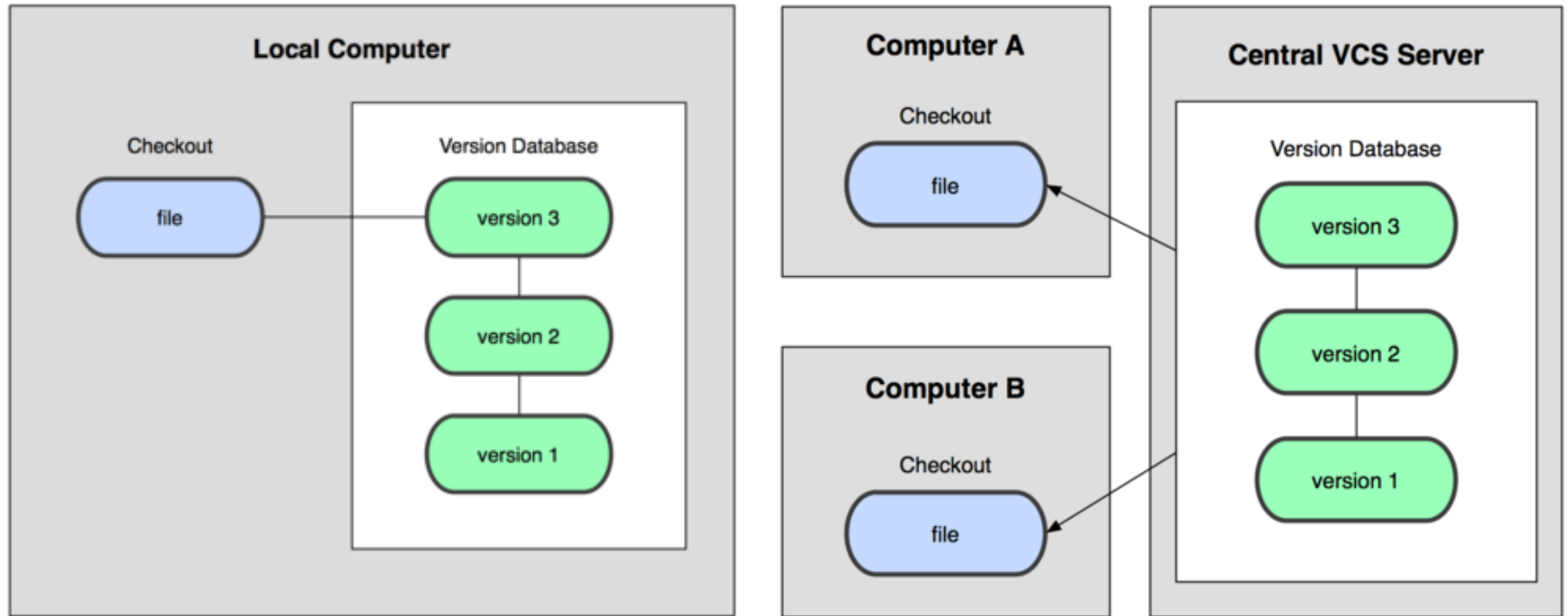
Git-it Workshop:  
<http://jlord.us/git-it/index.html>

Free at <http://git-scm.com/book>

# Version Control

- Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later
- .A Version Control System (VCS) allows you to:
  - revert files back to a previous state
  - revert the entire project back to a previous state
  - review changes made over time
  - see who last modified something that might be causing a problem and when
  - and much more.....

# Local and Central VCSs

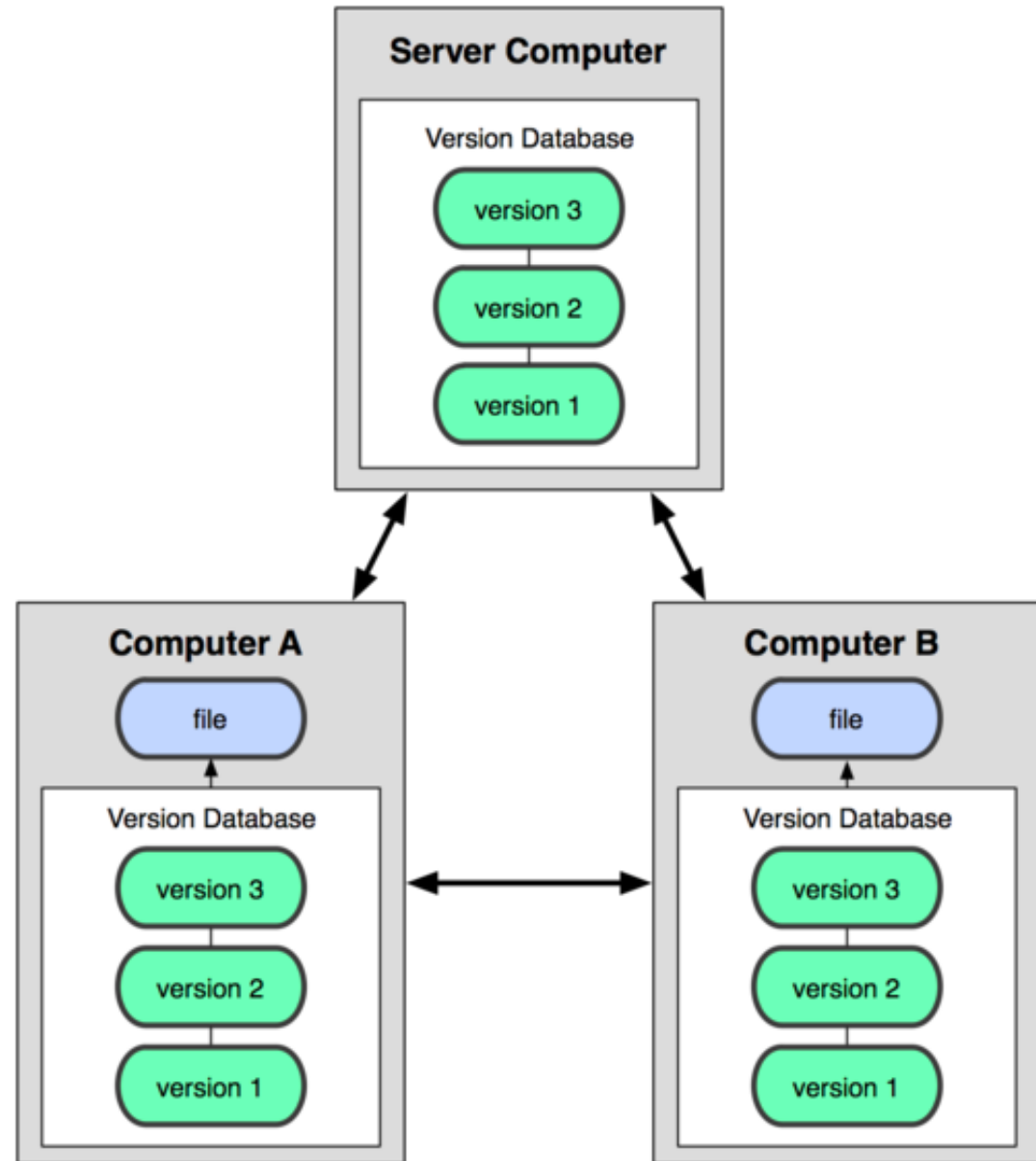


**Problem:** hard to collaborate

**Problem:** single point of failure of the centralized server, e.g. development disruption, losing code, etc.

# Distributed VCS

- Clients fully mirror the repository
- Every checkout is really a full backup of all the data
- If any server dies, any of the client repositories can be copied back up to the server to restore it
- Client can check in code w/o connecting to the server



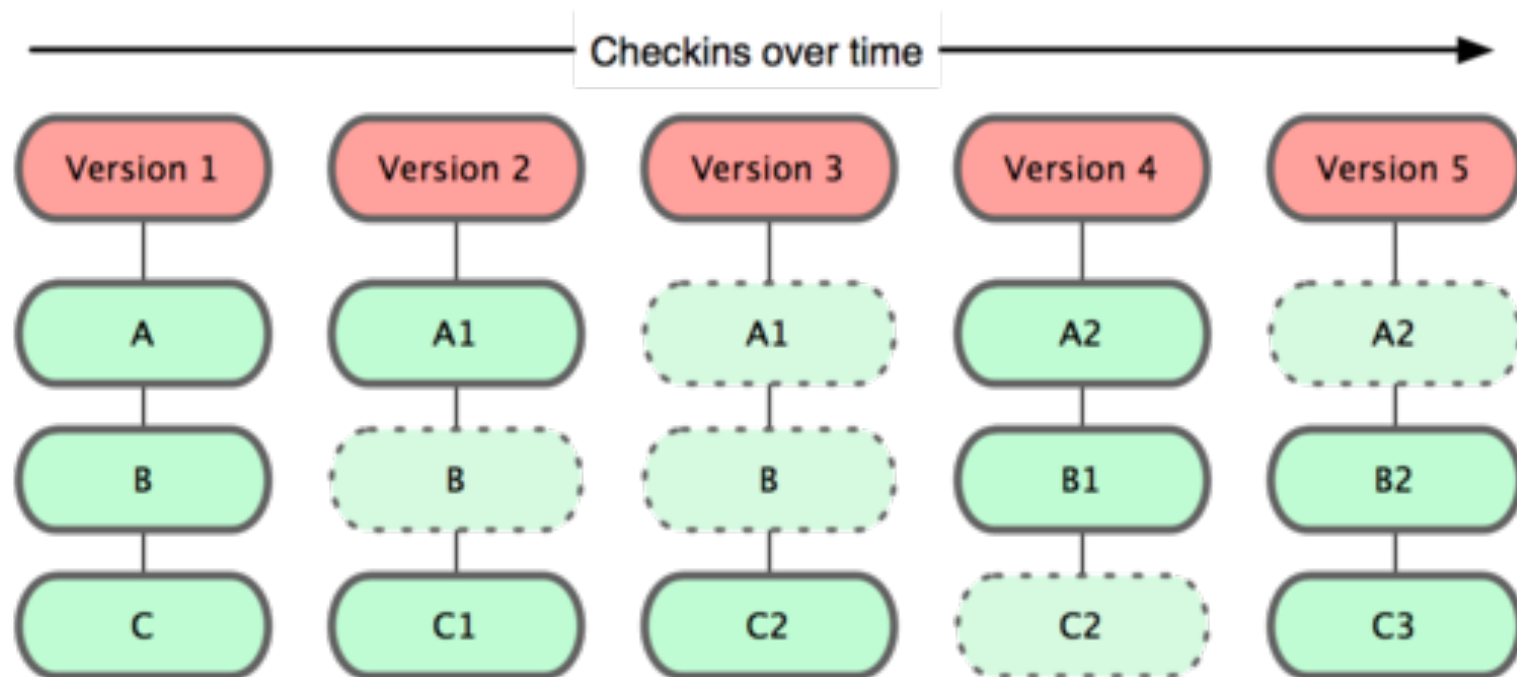
# Git

- Started in 2005
- Distributed VCS
- Very popular in developer communities
- Well-known git systems: github and bitbucket



# How Git store data

- Git stores data as snapshots of the project over time



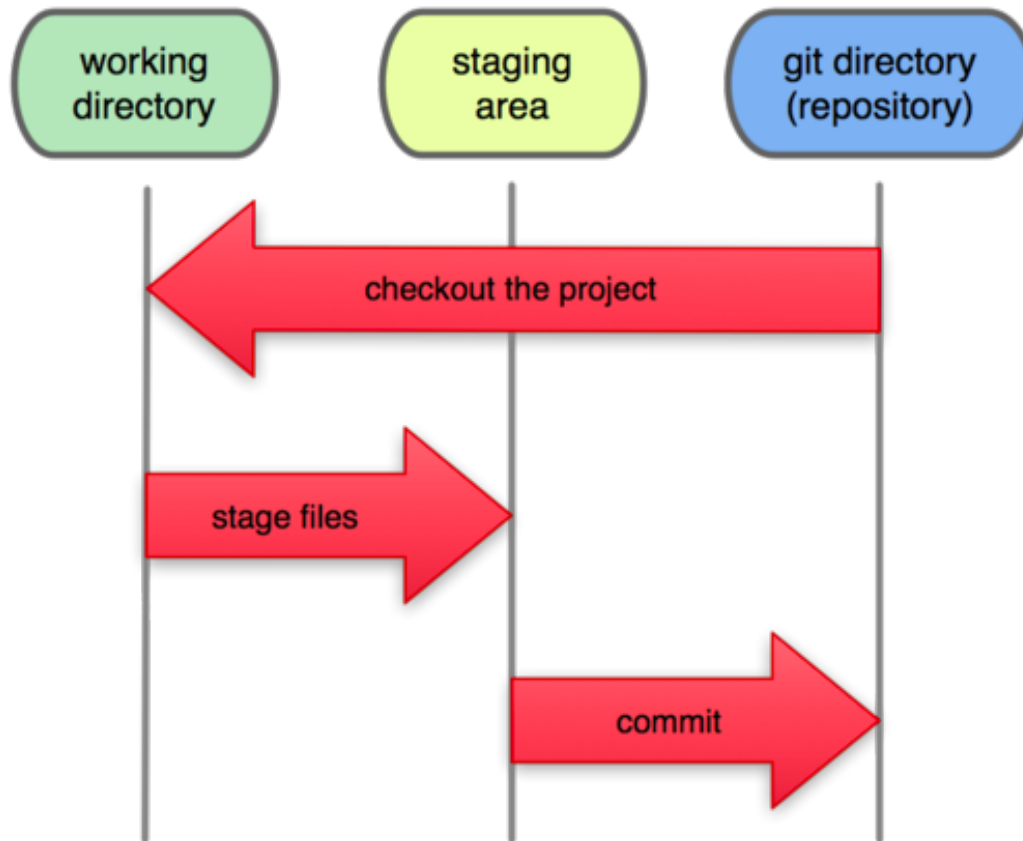
# Three States of Files in Git

- Git has three main states that your files can reside in: committed, modified, and staged.
- Committed means that the data is safely stored in your local database.
- Modified means that you have changed the file but have not committed it to your database yet.
- Staged means that you have marked a modified file in its current version to go into your next commit snapshot



# Three Main Status of a Git Project

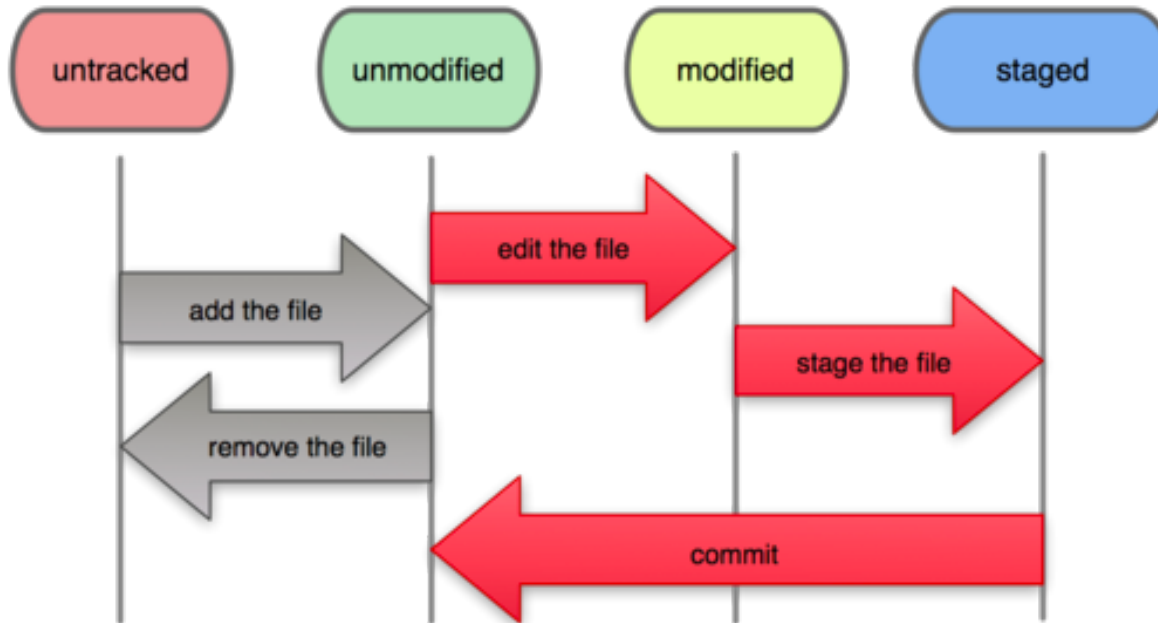
## Local Operations



***Command: git status***

# File Status

## File Status Lifecycle



- Untracked means that Git finds a file that didn't have in the previous snapshot (commit);
- Untracked files won't be included in commit snapshots until added to Git explicitly
- Git does this to prevent accidentally including generated binary files or other files that the developer did not mean to include.

# Work with Remotes

- Remote repositories are versions of your project that are hosted on the Internet
- Collaborating with others via managing these remote repositories and pushing and pulling data to and from them
  - Show remote repos: `git remote -v`
  - Add remote: `git remote add [shortname] [url]`
  - Pull/Push from/to remote:  
`git pull [remote-name] [branch-name]`  
`git push [remote-ame] [branch-name]`

# The Basic Git Workflow (no branching)

1. Build a repository locally (or clone a repo)
2. Modify files in your working directory.
3. Stage the files, adding snapshots of them to your staging area.
4. Do a commit, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory.
5. Push updates to the remote repo

# Git Basics Exercises

- We use a series of demos to illustrate Git basic operations
    - Setup an empty repository in github
    - Clone a repository
    - Create files, add files, commit
    - Push changes to remote repo
1. `git clone`
  2. `git status`
  3. `git add [filename]` or `git add *`
  4. `git commit -m "commit message"`
  5. Check commit history: `git log`
  6. `git pull/git push`

Questions?