Git for Version Control

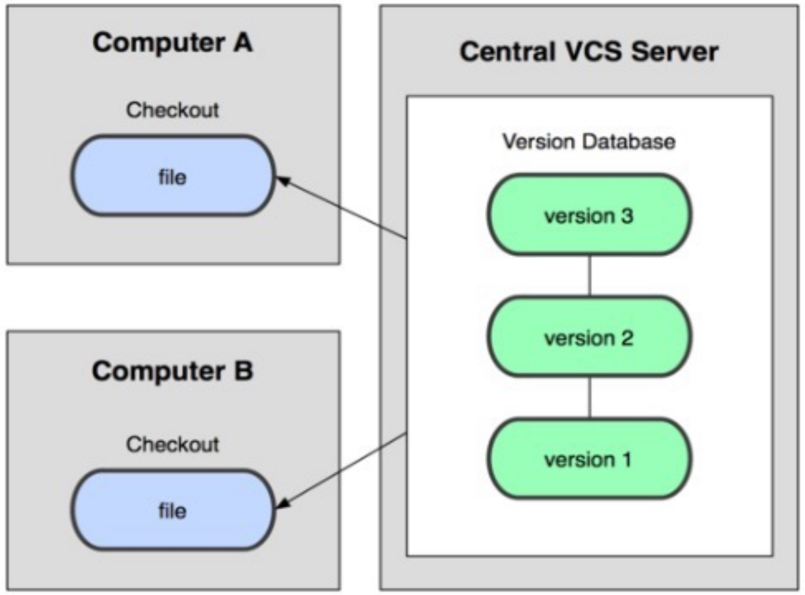
Dr. Katrin Schöning-Stierand

Version Control System - VCS

- Record changes to a file or set of files over time
- Access or revert to specific versions
- Simplifies collaborations

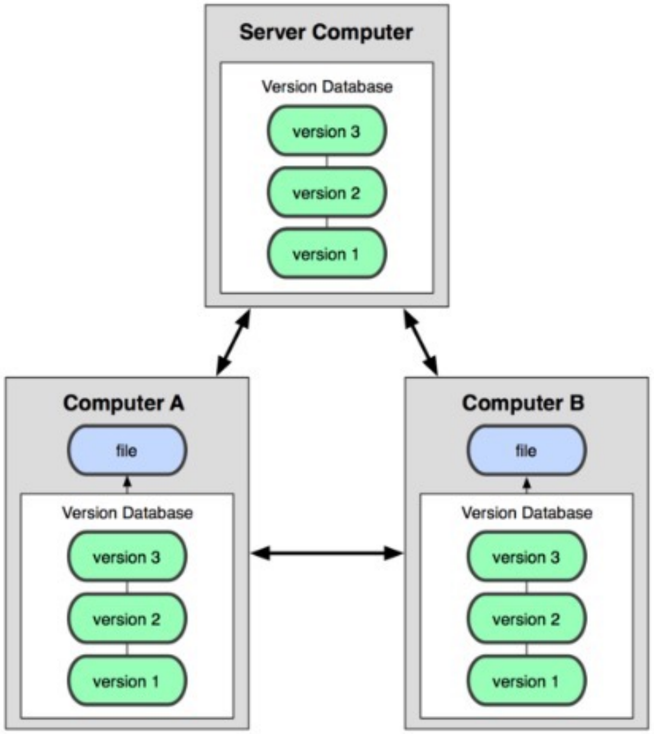
Centralized VCS

- In Subversion, CVS, Perforce, etc. A central server repository (repo) holds the "official copy" of the code
  - the server maintains the sole version history of the repo
- You make "checkouts" of it to your local copy
  - you make local modifications
  - your changes are not versioned
- When you're done, you "check in" back to the server
  - your check-in increments the repo's version



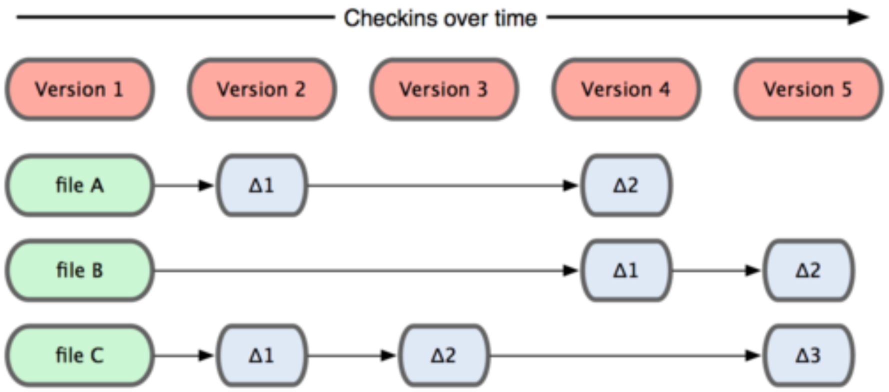
Centralized VCS

- In git, mercurial, etc., you don't "check out" from a central repo
  - you "clone" it and "pull" changes from it
- Your local repo is a complete copy of everything on the remote server
  - yours is "just as good" as theirs
- Many operations are local:
  - check - in/-out from local repo
  - commit changes to local repo
  - local repo keeps version history
- When you're ready, you can "push" changes back to the server

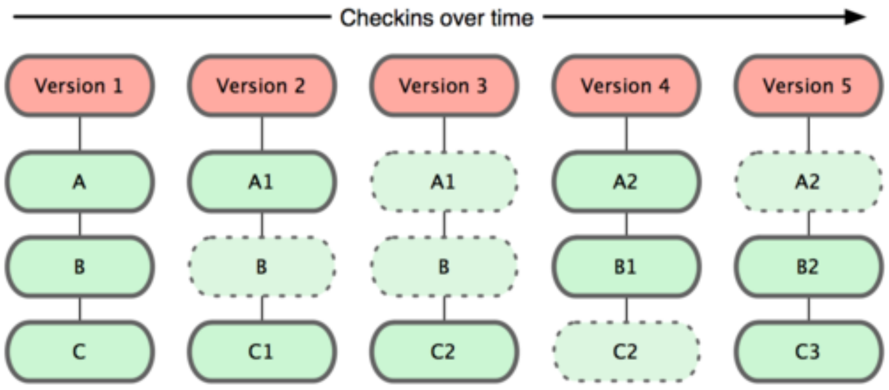


Git snapshots

- Centralized VCS like Subversion track version data on each individual file.

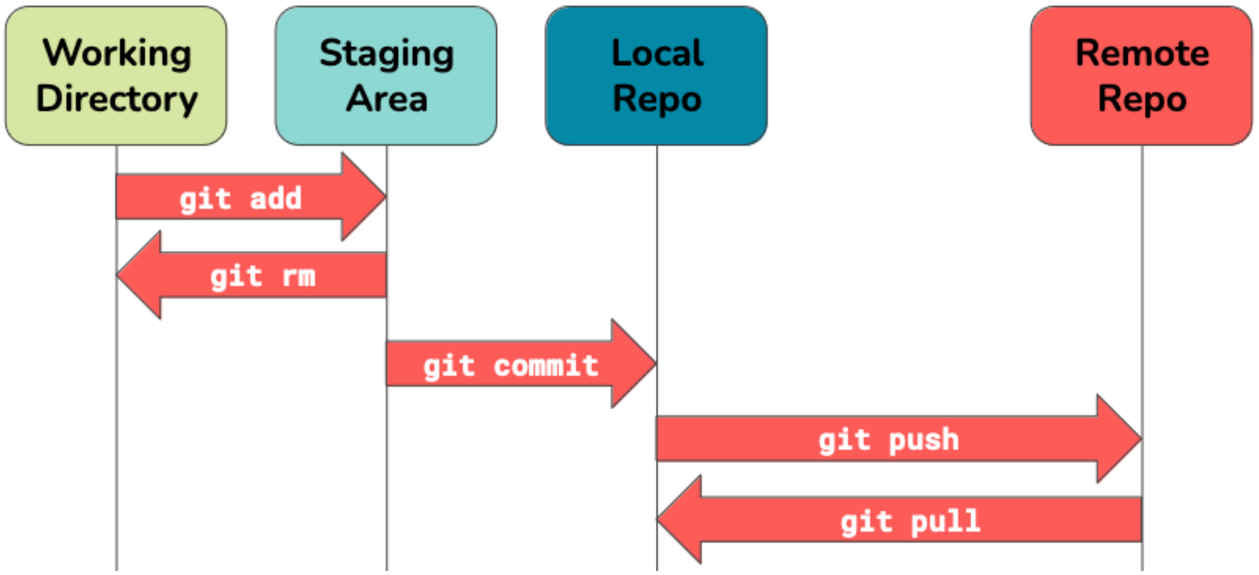


- Git keeps "snapshots" of the entire state of the project.
  - Each check-in version of the overall code has a copy of each file in it.
  - Some files change on a given check-in, and some do not.
  - More redundancy, but faster.



Standard Workflow in GIT

- **Modify** files in your working directory.
- **Stage** files, adding snapshots of them to your staging area.
- **Commit**, which takes the files in the staging area and stores that snapshot permanently in your Git directory.
- **Push**, which takes local changes to the remote repository
- **Pull**, which takes remote changes to the local repository



Installing/learning Git

- Git website: http://git-scm.com/
  - Free online book: http://git-scm.com/book
  - Reference page for Git: http://gitref.org/index.html
  - Git tutorial: http://schacon.github.com/git/gittutorial.html
  - Git for Computer Scientists
    - http://eagain.net/articles/git-for-computer-scientists/
- At command line: (where verb = config, add, commit, etc.)
  - git help verb

Git commands

command	description
`git clone` ***`url [dir]`***	copy a Git repository so you can add to it
`git add` ***`file`***	adds file contents to the staging area
`git commit`	records a snapshot of the staging area
`git status`	view the status of your files in the working directory and staging area
`git diff`	shows diff of what is staged and what is modified but unstaged
`git help` ***`[command]`***	get help info about a particular command
`git pull`	fetch from a remote repo and try to merge into the current branch
`git push`	push your new branches and data to a remote repository
others: `init, reset, branch, checkout, merge, log, tag`	