Git & Github

(The ‘No Frills’ version, adapted from http://rogerdudler.github.io/git-guide/)

Git is a Vcs

- A “version control system”
- Change tracking on files. "Backup" of versions of files, if you so choose.
- Enables multiple people to work on same code without too much headache.
- Git was initially designed and developed by Linus Torvalds for Linux kernel development

Git is distributed

- Every Git ‘working directory’ is a full-fledged repository with complete history and full version-tracking capabilities.
- A ‘working directory’ is just a copy on disk a ‘repository’
- A ‘repository’ is a code base that you want to collaborate on with others. (sometimes called a ‘repo’ for short)

Github is Git Hosting Service

- Github has generously donated an ‘organization’ to us.
- An organization is just a private site for us to share repositories as a group.
- Github will contain repos for each of homeworks, in-class code, etc..
- We will effectively download the code from Git to work on it, then we will upload the code back (We will do this through git commands.)

Basic Git Workflow

1. First, you ‘clone’ a repository from Github.
   (translation: make a local copy, you do this only once!)

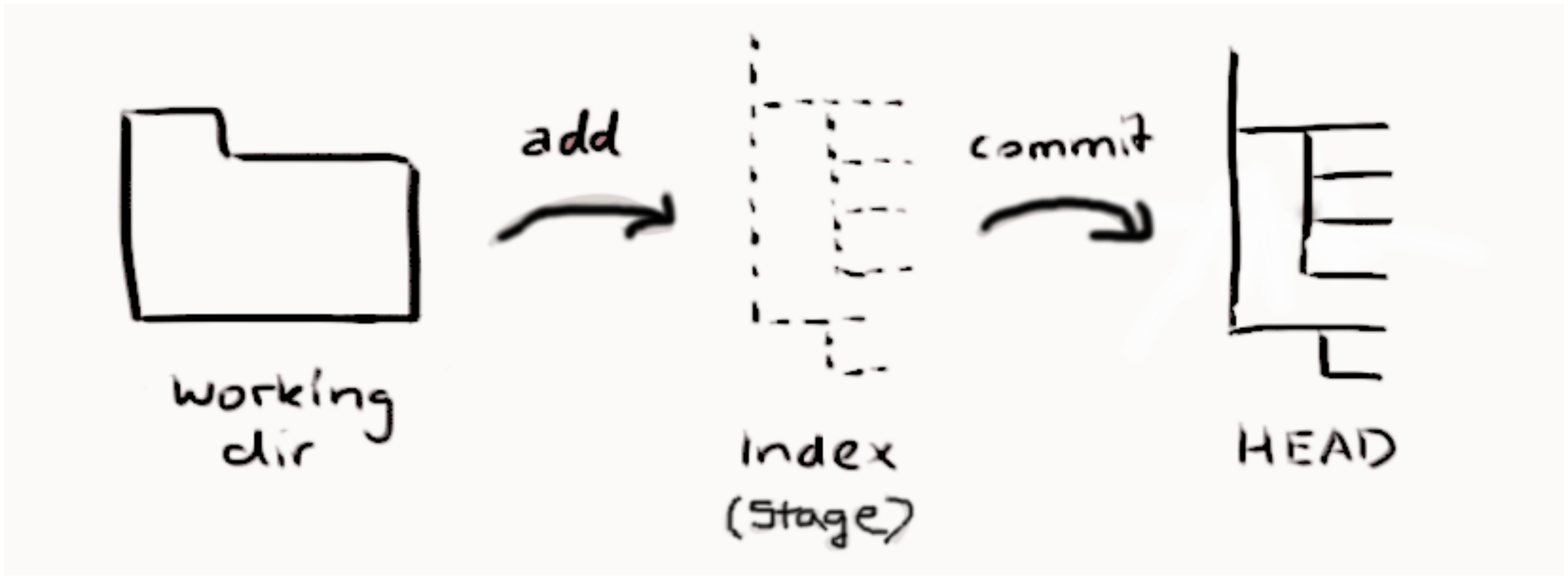
2. Next you ‘add’ new files and modify existing files.

3. Then you ‘commit’ those changes and additions.
   (translation: take a backup of that version)

4. Finally, you will ‘push’ that code to Github

Steps 1-3

- After you’ve cloned a repo…
    - your local repository consists of three "trees" maintained by git.
    - the first one is your ‘working directory’ which holds the actual files.
    - the second one is the Index which acts as a staging area
    - and finally the HEAD which points to the last commit you've made.



Steps 1-3

- You can propose changes (add it to the Index) using
  - git add .
- To actually commit these changes use
  - git commit -am "Commit message”
- Now the file is committed to the HEAD, but not in your remote repository yet.
- (Don’t break the build!! I.e. do not commit code that is known to be broken)

Step 4

- Your changes are now in the HEAD of your local working copy. To send those changes to your remote repository, execute
  - `git push origin master`

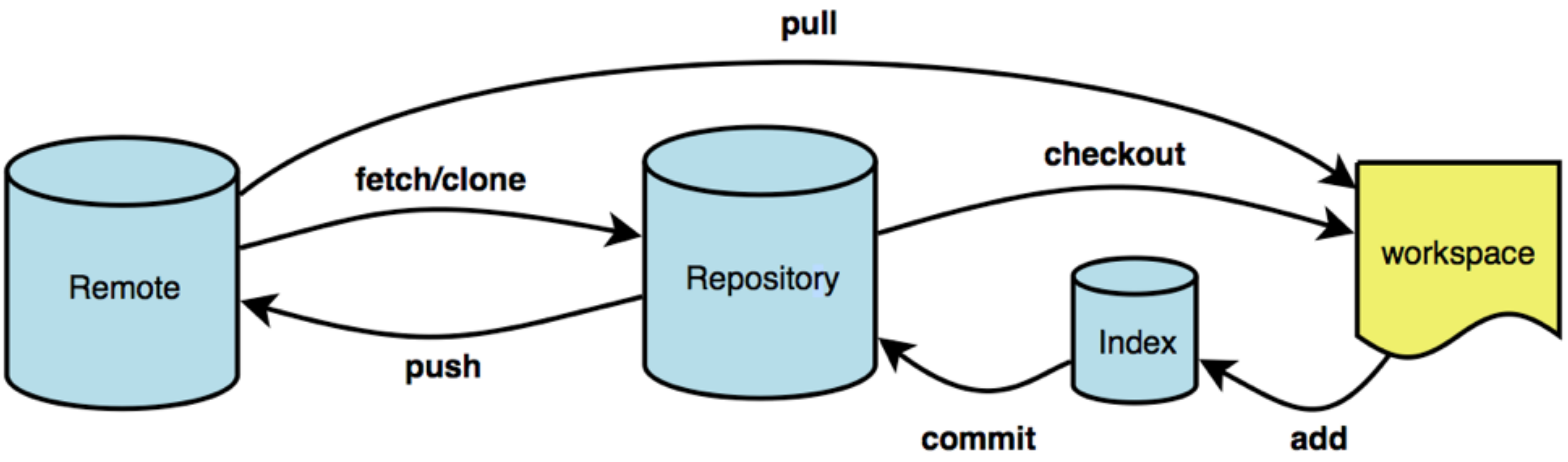
Step x.5

- Interleaved throughout that process you may want to see if your teammates have pushed anything.
- You can get their code by executing..
  - git pull origin master

Update & Merge

- git tries to auto-merge changes.
- This is not always possible and results in conflicts.
- You are responsible to merge those conflicts manually by editing the files shown by git.
- After changing, you need to mark them as merged with..
  - git add <filename>

Moreover…


pull
fetch/clone
checkout
Remote
Repository
workspace
Index
push
commit
add

Learning Resources

- Interactive tutorial on Git
  - https://try.github.io/levels/1/challenges/1
- Interactive tutorial on Git Branching
  - http://pcottle.github.io/learnGitBranching/