More Git

07-131 Great Practical Ideas in CS

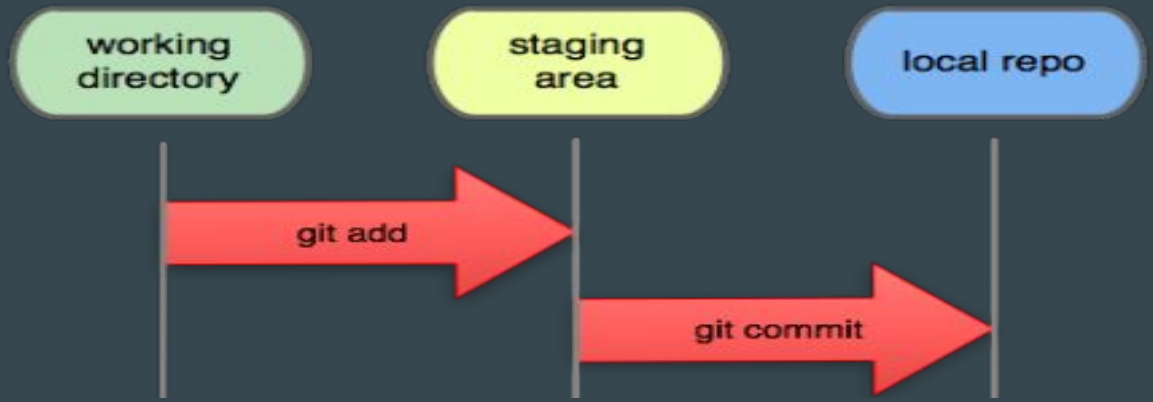
Instructors: Eduardo Feo-Flushing & Giselle Reis

Course website: https://web2.qatar.cmu.edu/cs/07131/

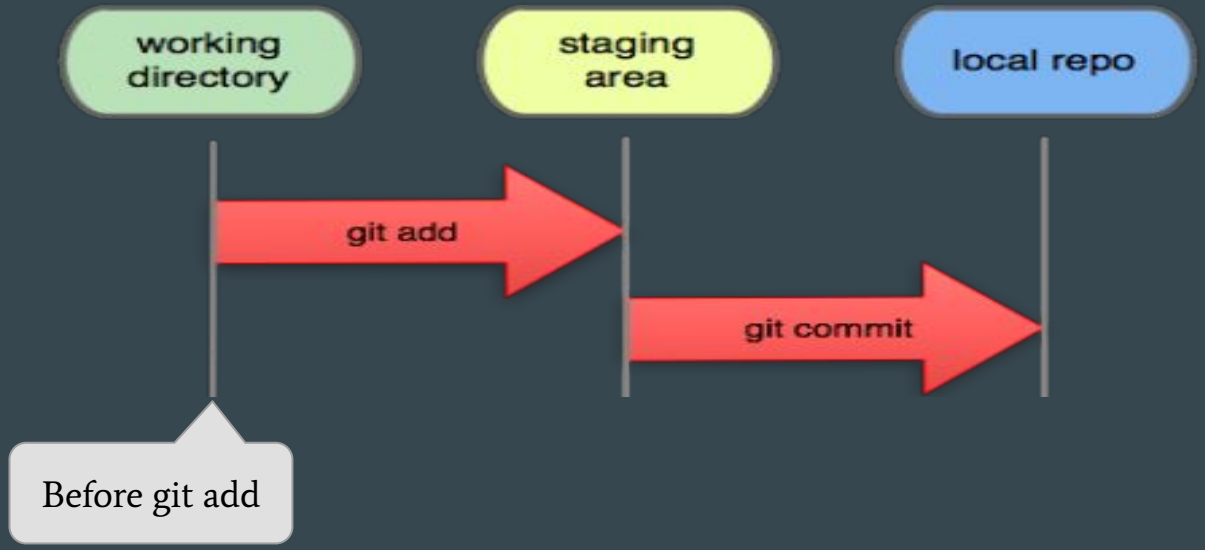
Git review

- `git status`
  - Use it often to find out what is happening!
- `git add`
  - Stages changes for commit (-p for staging partial changes of a file)
- `git commit`
  - Creates a commit with all staged changes
- `git branch <name>`
  - Creates a new branch from the current commit called name
- `git checkout <name>`
  - Switches to a branch called name

Git workflow (so far)

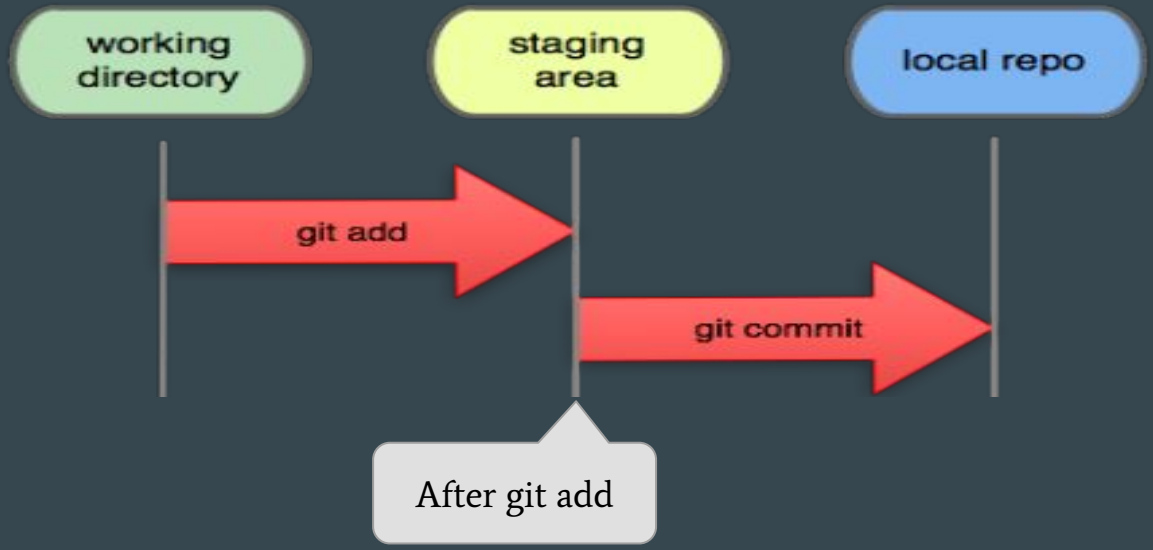


Undoing changes



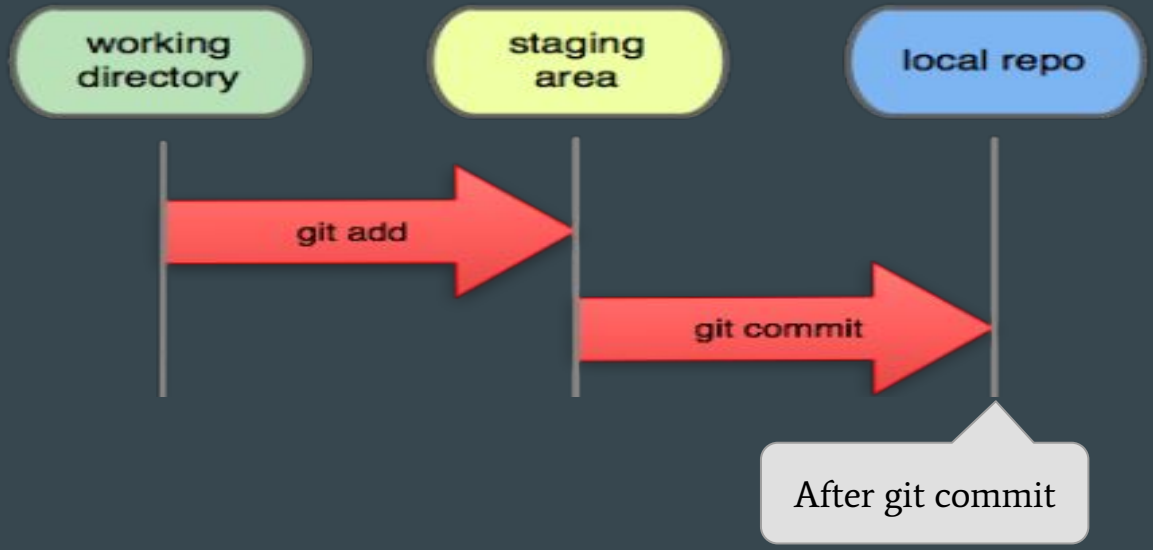

```
git restore <file>
```

Undoing changes



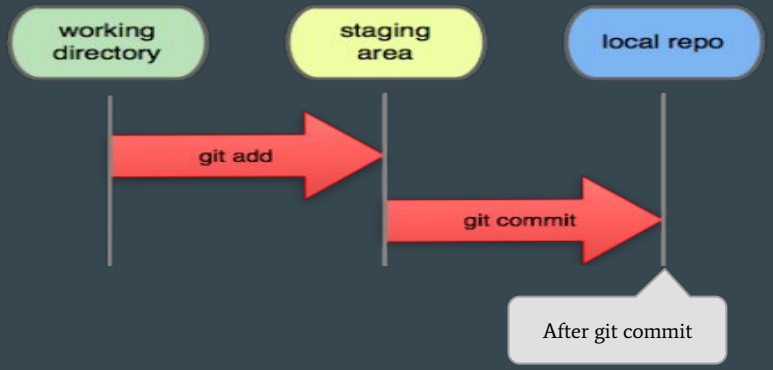

```
git restore --staged <file>
```

Undoing changes




```
git reset HEAD~1
```

Undoing changes



`git reset HEAD~1`

- Removes last commit. Use `HEAD~n` to remove last n commits
- Different from `git revert <commit hash>`
- Changes from the commit become unstaged

Saving changes for later

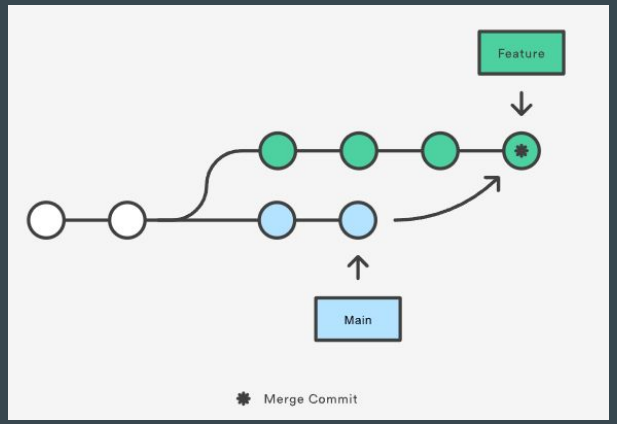
Sometimes we need to pull changes from a server, or merge branches, but we have local (staged or unstaged) changes that we are not ready to commit.

You can save the changes locally, do whatever you need to do at the HEAD of the repo, and recover the changes to continue your work.

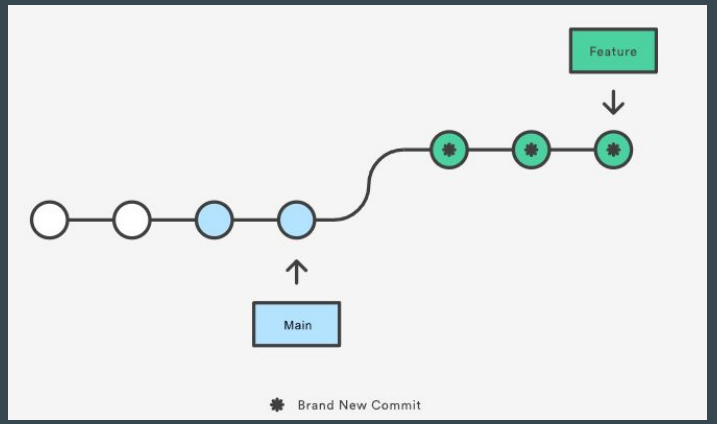
- `git stash` – saves your local uncommitted changes on a stack
- `git stash pop` – recovers the last set of saved changes on the stack
- `git stash list` – lists the entries on the stash stack

Rebasing branches

We have seen how to merge branches. If we do not want to see *merge commits*, we can `rebase` instead.




```
git checkout feature
git merge main
```




```
git checkout feature
git rebase main
```

Fixing conflicts

Rebasing or merging may result in conflicts if two different commits change the same region of the file.

Conflicted files will have conflict regions indicated by:

```
<<<<<<< Your changes
the content on your branch
=======
the content on their branch
>>>>>>> Their changes
```

To fix them, edit the files to the state you would like them to be, then add and commit the changes.

I like to use `git mergetool` to fix conflicts.

Interactive rebase

If you need to organize your commits before pushing or merging them, you can do so by using interactive rebase:

```
git rebase -i HEAD~n
```

will give you a list with the last n commits and instructions how to modify them.

You can use interactive rebase to rebase one branch onto another too!

Github

git repositories in the cloud

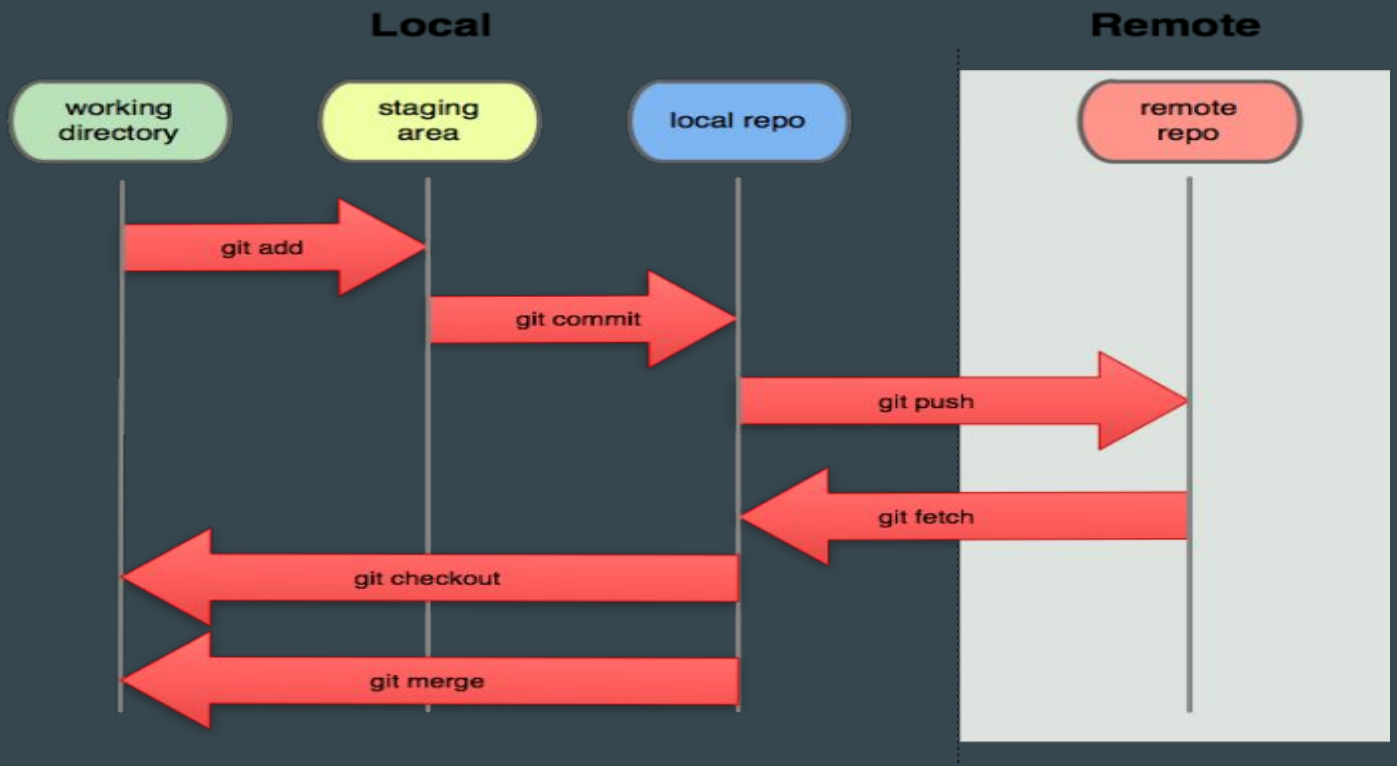
Github

- Service for hosting git repositories online
- Backup for your git repositories
- Version controlled dropbox :)
- Very useful for collaboration (go contribute to some open source projects!!)
- Alternatives to github
  - BitBucket
  - GitLab
  - SourceForge
  - Your own server!

Remotes

- Copies of your git repository that are stored elsewhere are called remotes
- The name for the default remote is `origin`
- A repo can be linked to multiple remotes
- Remote commands:
  - `git remote -v` – see all remotes
  - `git remote add <name> <url>` – adds a new remote pointing to url called name
- push and pull commands communicate with a remote
  - `git pull <remote name/remote branch> <local branch>` – brings commits from remote branch into local branch (if omitted, uses default remote – origin – and current branch)
  - `git push <remote name> <remote branch>` – sends local commits to remote branch (if omitted, uses default remote and *upstream* branch)

Remotes


Local
Remote
working directory
staging area
local repo
remote repo
git add
git commit
git push
git fetch
git checkout
git merge

Github only concepts

- Fork
  - A copy of a repository in another user's space or organization.
  - The connection between those is done by github.
- Issues
  - List of bugs, feature requests, improvements, etc, in a project.
  - Good place to start when contributing to open source projects!
  - Issues can be closed automatically if your commit message includes the text `Fixes #n` (or any of: close, closes, closed, fix, fixes, fixed, resolve, resolves, resolved)
- Pull requests
  - Asking owners of the forked repository to use commits from your repository

It's Collab time! (Attention, this lab has two parts!)