Git Walk-through

## What is Git

Git is an example of a DVCS (hence Distributed Version Control System). Rather than have only one single place for the full version history of the software as is common in once-popular version control systems like CVS or Subversion, in Git, every developer's working copy of the code is also a repository that can contain the full history of all changes.

## Git Workflow

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.



## What is a pull request ?

Pull requests let you tell others about changes you've pushed to a GitHub repository. Once a pull request is sent, interested parties can review the set of changes, discuss potential modifications, and even push follow-up commits if necessary.

## Pre-pull Requests

## Squash

With git it’s possible to squash previous commits into one. This is a great way to group certain changes together before sharing them with others. ~ Here’s how to squash some commits into one. Let’s say this is your current git log.

## Stash/Pop

Stashing takes the dirty state of your working directory — that is, your modified tracked files and staged changes — and saves it on a stack of unfinished changes that you can reapply at any time.

## Tagging

A tag represents a version of a particular branch at a moment in time. Usually you'll tag a particular version so that you can recreate it, e.g., this is the sprint 22 release.

## Issues

To close an issue in the same repository, use one of the keywords in the list below followed by a reference to the issue number in the commit message. For example, a commit message with Fixes #45 will close issue 45 in that repository once the commit is merged into the default branch.

* close
* closes
* closed
* fix
* fixes
* fixed
* resolve
* resolves
* resolved