Software Configuration Management

Definition
History
Current, popular SCM options
GUIs
GUIs
Play Time!
A realistic use case
Other topics/terms

SCM - Definition

VCS – Version Control System

SCM - Definition

VCS – Version Control System

VCS IS:

- the place to store (and version) code
- the place to store build scripts
- the place to store automated testing scripts

SCM - Definition

VCS – Version Control System

VCS IS:

- the place to store (and version) code
- the place to store build scripts
- the place to store automated testing scripts

VCS IS NOT:

- the place to store dependencies
- the place to store documentation
  - except READMEs
- the place to store manual test cases & results

VCS - History

Story Time!

VCS - History

Story Time!

Simple, aka “archaic”

- Shared drive with locking capability
- Examples: RCS, VSS

Enterprise, aka “monolithic”

- Lots of structure and management
- Examples: IBM/Rational Clearcase

VCS - History

Subversion

- *Centralized* VCS
- Repo Structure:
  - *trunk*
  - *branches*
  - *tags*
- Branching copies entire directory structure
- Sync Merging
- Code Review by changeset

VCS - History

Subversion

- *Centralized* VCS
- Repo Structure:
  - *trunk*
  - *branches*
  - *tags*
- Branching copies entire directory structure
- Sync Merging

What if I'm working remotely?

Who's in control of the trunk?

VCS - Git

Distributed VCS

- Everyone has a copy of the entire repository
- Git tracks the deltas
- Remote repositories
- Code Review by Pull Request (PR)

Terminology

- Add
- Commit
- Fetch
- Push
- Pull

VCS – Git Branching

Repo structure

› http://nvie.com/posts/a-successful-git-branching-model/

develop/ - main branch of development and primary integration point

features/ - all feature development

release/ - created from develop/ when release is ready for user testing

master/ - post-release, tagged version of the code

hotfix/ - any production-level bugs that need to be resolved against master/ and develop/

VCS - Mercurial (Hg)

Another *Distributed* VCS

Supposedly easier learn, with clearer commands that mirror Subversion

Locks down the history

Which is better?

VCS – Mercurial (Hg)

Another *Distributed* VCS

Supposedly easier learn, with clearer commands that mirror Subversion

Locks down the history

Which is better?

- Probably doesn't matter with today's GUI tools

VCS - GUI Tools

Atlassian's SourceTree

Git Extensions

TortoiseGit

IDE Integration

- Eclipse
- IntelliJ
- Xcode
- Visual Studio

SCM - Platforms

Self-hosted *vs.* Cloud Hosted

- Security
- Cost
- Integrations

Cloud Hosted

- Github
- BitBucket
- VSTS

Git – Play Time (The Basics)

git init
git checkout –b master
git status

Git – Play Time (The Basics)

git init

git checkout –b master

git status

git checkout –b develop

Create a file README.md in your directory

› # This is my README

Git – Play Time (The Basics)

git init

git checkout –b master

git status

git checkout –b develop

Create a file README.md in your directory

› # This is my README

git status

git add README.md

git status

Git – Play Time (The Basics)

git init

git checkout –b master

git status

git checkout –b develop

Create a file README.md in your directory

› # This is my README

git status

git add README.md

git status

git commit –m "my first commit"

git log

Git – Play Time (The Basics)

git checkout –b features/my-first-feature

git status

Edit your file

git add README.md

git commit –m 'edit from my branch'

git log

git show <commit #>

Git – Play Time (The Basics)

git checkout develop
git log
git merge my-first-feature
git log

Git – Play Time (The Basics)

git checkout develop
git log
git merge my-first-feature
git log
git checkout –b master
git tag -a v1.0.0 -m 'Initial Release 1.0‘
git tag
git show v1.0.0

Git – Play Time (The Basics)

git branch –d my-first-feature
git branch

Git - Remotes

A central repository is still needed

- Multiple remotes

git remote add origin https://github.com/joshkgold/ud-sample-repo

git push origin master

- Push local code to central repo

Edit the file in github

git pull origin master

Git - Remotes

git log

› See change from central repo?

git pull origin master

git checkout develop

git merge master

git push –u origin develop

Git – A Real-World example

Create yet another feature branch

Edit the README.md file on the same line

git push –u origin <branch name>

Github account

- Create another Pull Request.
- Conflict? How do we resolve?

Git – Other Helpful Commands

"I need to just start over"

- *reset*
- *git reset --soft HEAD~*
  - *Move the commit pointer to the specified last commit*
- *git reset --mixed HEAD~*
  - *Move the commit pointer and undo the commit*
- *git reset --hard HEAD~*
  - *aka UNDO command*

https://git-scm.com/blog - Reset Demystified

Git – Other Helpful Commands

"Let's keep this for later"

- *stash*
- *git stash*
  - *Saving current work in Working Copy*
- *git stash list*
  - *List all stashed code*
- *git stash apply stash@{2}*
- *Revive the 2nd stashed set of code*

https://git-scm.com/book/en/v1/Git-Tools-Stashing

Git – Other Commands/Terms

Clone
- Copy down a repository

Cherry-picking
- Merging a specific commit from another branch

Fast-Forward
- When Git realizes it can reduce the # of nodes in the history and shorten them

Revert
- Bring a previous commit back to the front

SCM

- *https://www.atlassian.com/git/*
- *https://git-scm.com/doc*

SCM

- *https://www.atlassian.com/git/*
- *https://git-scm.com/doc*

- *Questions?*