Software Engineering Fundamentals:  
Source Code Management

short line

# What Is It?

Source code management or SCM is the process of managing a software products code base. SCM is a process that has been around since the inception of software development. It is a way for developers to collaborate, sharing one code base and ensuring that all developers have the same source code.

Throughout the history of software engineering there has always been a need for source code management. As projects grow and more engineers start to contribute, the code base becomes more difficult to manage.

# How is it Applied

Typically source code management is accomplished with a version control system. There are three main types of version control systems, Local, Centralized and Distributed.

## Local Version Control System

Local VCS involves using the local file system to manage code base versions. This system typically uses different folders to manage code and can lead to issues such as using the wrong version of code.

## Centralized Version Control System

Centralized VCS involves using a repository that is hosted on a server with each developer being a client of that server. Centralized VCS is an improvement over Local VCS due to the code base being stored in a central location. Versions of code are managed by software which removes the control from the developer. Although this type of VCS is an improvement over local vcs, it does have drawbacks.

## Distributed Version Control System

Distributed VCS is similar to centralized in that the code base is managed by a repository system on a server. Unlike centralized, the repository is also stored on the local system of each client. When changes are made to the code base updates are pushed to the repository and then synced to each client. It is important to note that the distributed system address many of the drawbacks of centralized systems, but they also have drawbacks as well.

# Version Control Implementations

## GIT

GIT is a distributed version control system. It is the most popular systems available. There are a few places that offer the remote server hosting.

* <https://github.com/>
* <https://about.gitlab.com/>
* [https://bitbucket.org](https://bitbucket.org/)

## Apache Subversion

Subversion is a centralized version control system that is widely used.

* <http://svnbook.red-bean.com/en/1.7/svn.intro.whatis.html>

# Version Control Best Practice

## Git Flow

GIT flow is one of the more common GIT best practice methods. It is implemented through the branching system of GIT. Typically there are three common branches: Master, UAT, and Development. There are also sub branches that are used for development, these include feature, bug, and hotfix branches.

### Master Branch

The master branch is where the production code for the project is stored. This is the code that is used to build the system that the business customers use, that employees in the company use when working. It is typically a slow moving environment, changes are thoroughly tested on previous environments. This branch is typically behind the UAT and Development branches in terms of commits.

### UAT Branch

The UAT branch is for user acceptance testing. Typically this is a small group of users that are both technical and from the business group. All new features and bug fixes are typically tested in this environment before being promoted to the production environment.

### Development Branch

The development branch is for live development. It is where new features and bugs are integration tested. This branch will be the fastest moving environment and will often be a few commits ahead of UAT and Master.

### Feature Branches

Feature branches are based off of development and is where new functionality is developed. The primary purpose of a feature branch is to develop new features and test the functionality without affecting development code.

### Bug Branches

Similar to feature branches, bug branches are based from development but are used to fix bugs that have been identified through testing.

### Hotfix Branches

Hotfix branches are similar to bug branches. They are used to fix bugs, but these bugs are typically production breaking issues. These branches will be based on the master branch and back commited into UAT and Development once the bug is fixed.

## Subversion

# Recommended Tools

* <https://git-scm.com/downloads>

# Resources

* <https://git-scm.com/>
* <https://git-scm.com/doc>
* <https://git-scm.com/book/en/v2>
* <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>
* <https://svn.apache.org/repos/asf/subversion/trunk/doc/user/svn-best-practices.html>

short dash