Nicolas Gonzalez

May 31, 2023

CIS233DA 17130

Professor McHardy

Source Control Tools

**Git**: This source control tool is the most widely used today and rightfully so. Git uses `distributed version control which means that the repositories can be cloned and used offline completely. Another feature of Git is that branching and merging is lightweight and easy to do. This allows for an easier workflow for teams who lack the proper coordination required for more difficult projects. Git also features staging which allows for developers to select what changes are included in their next commit. Since Git is the most popular programming language there is an amazing amount of resources to assist developers in troubleshooting and learning its features. A downside to Git is that there are limitations to renaming folders and files. Git primarily tracks file content changes rather than individual files or folders. Another downside is that very large repositories with numerous files or a long history may experience some performance degradation.

**Subversion**: One of the more popular source control tools, SVN is still actively being used by many. SVN is centralized which means that one single repository holds all of the version history. In contrast, Git allows each user to copy the repo completely allowing for offline use and makes branching and merging easier. Not being able to work effectively offline is one of Subversions biggest downsides.

**Mercurial**: This version control tool is well known for being powerful and it’s easy to use interface. Mercurial offers customizability using extensions and plugins which can do things such as handle larger files and code review. It also features a web interface much like GitHub that allows for version control on a web browser. A downside to Mercurial is that since it isn’t as popular as Git, there is a lack of support to troubleshoot code. Another downside is that extensions can cause some dependencies for the project.

**Perforce**: Perforce is a centralized source control tool that offers a range of unique features that differentiate it from the other tools. It offers unique features such as atomic commits, heavy branches called “streams”, and high scalability. Atomic commits in Perforce means that either all changes to the project are accepted simultaneously or none of them are. Used to simplify branches in any project, Streams help increase efficiency in any given project. A downside to Perforce is that it’s a centralized version control system just like SVN. Another downside is that Perforce requires you setup a dedicated server which will use your own resources.

**Team Foundation Server**: Rebranded to Azure DevOps in 2020, TFS was created in 2005 by Microsoft. This source control tool is offered by Microsoft and offers more than just version control. Something that makes TFS unique is that it includes build and release management features that allow teams to automate the build, testing, and deployment processes. Another feature is that it can be centralized or distributed, which allows for more diversity in how a team prefers to manage the versions of their project. TFS also features extensions much like Mercurial, which allows for even more customization and features. A downside to TFS is that, due to its amazing customizability, there is a steep learning curve. Also, another downside is that there is a licensing cost to use it.