Hope Lepeshko

CSCI-3327-001

GitHub Essay

Git is a distributed version control system also known as VCH. In plain English Git is basically a super powerful tool software developers can use to save their files. One of the best parts about Git is that it allows developers to collaborate with each other while storing a history logbook and commit log with details of changes made to a repository. A repository, often called repo for short, is like a digital file folder in which every file related to a project is stored. Inside the repository is where you will find all project folders including code, code documentation, screenshots of code output, and anything else a software developer might deem meaningful to the project. The repository is also where the detailed log of every change made to the files is stored. This is especially useful if there was a mistake made. Developers can easily restore their code to its state prior to the mistake being made. This ability to correct mistakes is why it is important for developers to follow guidelines when using Git to manage source code. This set of guidelines is often reffered to as Git workflow. Git workflow gives developers guidelines to follow when changes are made, when code is reviewed, and pretty much any other step in Git. This allows for consistency which helps developers work together more seamlessly. Git workflow can be broken down into a few different areas including feature branches, committing changes, pull requests, merge requests, code review, fetching, pulling, and resolving conflicts.

The concept of commits refers to a saved state of your code. This means the code which existed before you made your changes to it, is still there. Which as mentioned previously allows developers to track down problems and come up with solutions. More quickly by restoring their code to a previous state. Every sing commit Git is unique, similar to DNA, or a human’s fingerprint. This allows large teams of developers to hold contributor accountable for any mistakes they may make, because we can always track down exactly who made which changes. There are no scape goats, or finger pointing when it comes to commits as even the time can be traced for every change made.

Developers will often create separate branches in repository for different feature, or different debug/fixes. The beauty of a branch is that it can remain separate from the main line of code until the developers make the decision to merge it back together with the master branch. This is a very useful aspect because many developers can work on the same project simultaneously without fear of getting in each other’s way. But just because developers can work separately at the same time does not mean that all conflict is avoided. Unfortunately merge conflicts can occur and cause issues. Merge conflicts happen when two branches have changes which clash and cannot be merged seamlessly. Merge conflicts often have to be resolved manually.

Pushes and pulls are another aspect of Git which makes collaborating with team members that much easier by keeping everyone’s code in sync. A push will send your local commits to a remote repository and a pull fetches the changes from a remote repository to your local machine.

In summary Git is one of the most valuable tolls modern developers have in their tool belt. Git provides developers with a powerful and efficient foundation for tracking changes, collaborating with team members and managing projects of any size big or small.