Anthony D’Alessandro

Probability and Applied Statistics – Git Paper

In the professional field of technology, software is at the core of almost all business operations. With the countless number of businesses and engineers that work on software, there is a need for an efficient and reliable way of saving, sharing, and recovering code. This is where version control systems are used to alleviate such problems and streamline the software development process.

Version control systems such as Git, are simply tools or systems that help track any changes made in various forms of documents, especially source code. Git is typically utilized by *cloning* a central repository to store code for a project in conjunction with the central repository that is stored non-locally in places such as GitHub. As programmers make changes and test their code, they have the ability to *commit* changes to their local repository where any edits are then saved. In the case that the developer’s changes are approved and ready to be officially added to the project, they can *push* their changes from the local repository to the central repository where the changes will be saved in the *main branch*.

However, before pushing changes to the central repository, there are some conditions that the developer must meet to ensure the integrity of the project. First, they should check the history of the central commits to make sure they are completely up to date on all changes. Then, the developer should confirm that their changes do not cause any conflict with the most recent commits. One useful feature built into Git is the ability to detect errors in which it will not allow a user to push any changes if their commits do not perfectly align with the current history of commits in the central repository. This helps reduce any overwrite errors and will essentially alert the developer that they need to *pull* any updates into their local repository and start the process over again.

When developers try to push changes that do not align with the central repository, a merge conflict may arise when attempting to merge branches that contain conflicting commits. Merge conflicts may arise before a merge even starts which happens when there are conflicts in the working directory. In addition, merge conflicts may occur during the merge process when there are conflicting commits in the local branch and one of the branches that are attempting to merge. Git provides developers with some useful commands for resolving merge conflicts such as *status, diff, checkout, abort, and reset.* The reset command is especially useful because it allows the user to change any conflicting files back to working conditions which may help identify where the issue occurred.

Git is a flexible version control system that is used very differently between companies or individuals. It provides the ability to share, edit, and recover files that may otherwise be lost or take considerably longer to complete. At its core, Git allows developers and technology companies to collaborate seamlessly on projects that are integral to society and help push people forward into a better future.