Collaboration is an important idea to understand and utilize in any field. In the field of Computer Science, the idea of collaboration extends beyond coworking in an office or coffee bar. Our understanding of collaboration is given form in cloud-based Version Control Systems. These systems allow individuals and teams in organizations the ability to create, modify, troubleshoot, and test code in any environment, safely and efficiently. Focusing on the services provided by the platform Github, this short essay will expand on important uses of the system and how it impacts our day-to-day workflow, allowing greater opportunities for programmers of any skill level.

The topics I wish to expand upon include cloning a repository, basing a project on specific branches of a repository, commits, pushing to a branch, checking out a branch, and merging. All of these uses of VCS have their own benefits, and together, form the basis of what I understand as the greatest boon to a Computer Scientist’s daily workflow. In briefly detailing each of these uses of VCS, any interested reader will be able to understand, and possibly implement, these skills to supplement their own use of services such as Github.

As part of a team in an organization, members may be tasked with different goals on a project. Utilizing Github allows these individuals to simultaneously work on an identical version of a project, based in the same foundation from a shared repository. Cloning is the process of creating a local project from a remote source. By cloning a project to perform some work on a program, individuals can work without their efforts being altered by other changes to the project by their teammates. This also allows these individuals to work in their desired environment.

After writing, testing, and reporting on their work, the next step is to commit and push the work to a branch of the repository. These actions are what allow teams to submit their work for review, and for reviewers to understand and analyze their changes before finalizing them and adding them to the project. Each commit is tracked with a separate ID and explains who submitted it and why, along with a summary of the changes added or made. By summarizing work well in the notes of a commit, teams can more accurately track what works and what doesn’t, aiding in their ability to collaborate.

Cloning may be confused with checking out a branch. While the commands allow users to access the data from repositories, they do so in different ways. As stated previously, git clone bases a new project on a remote repository. Git checkout allows a user to switch between branches that have been cloned locally. This gives further flexibility to individuals when testing and writing for different goals and with different dependencies in play. By utilizing this command, individuals can quickly and efficiently test in multiple environments locally to collaborate with teammates on shared goals.

Finally, merging allows users to compare and base a version of the project on both the repository source and their local source. When conducting a merge, the user will be prompted to compare the files from both sources. If file versions conflict, a merge conflict occurs. The user will be shown each version of files and must choose a version to pass to their environment. This conflict may negatively engage a misunderstood user. However, the merge conflict is important because it displays differences in concurrent work that may have been committed by several users, or the same user across different platforms, like a work machine and a personal device. By managing merge conflicts, teams can stay more accurately abreast of the most up-to-date work on relevant projects.

In detailing several normal, but important, scenarios and topics that can make up or encompass an individual’s team-based workflow, I hope to have illustrated the importance of learning and understanding Version Control Systems. In applying the use of these skills through Github to a work environment, any able user can improve their proficiency in coworking and become a greater asset to any development or analysis team.