Dominic Zelinsky
Professor Byron Hoy
CSCI-3327-001
2/5/2025

Git is a version control system. A version control software or VCS is a software that is used to provide the history of changes made to a file while allowing users to revert to previous versions on record. This allows for more in depth collaboration between multiple people by being able to track all the various edits made to the files.

Git is the tool, Git workflow provides a more standardized way in which the tool is used in professional environments. If everyone is able to push edits to said files within git, it can get messy very quickly. The Git workflow is essentially the flow in which changes would be made towards a file to ensure the team stays on track and prevent errors within said file. There are a few different popular work flows with git, such as Centralized Workflow, Gitflow Workflow, and Forking workflow. Gitflow Workflow creates a branching model that is designed around the project release. It accomplishes this by using different branches that have specific roles that determine how they interact. Forking Workflow is where the developers have two repositories. A public one which is shared amongst each other and a local private one where they do the majority of their work. The one that is most popular and likely a foundation that other workflows are based on is Centralized Workflow.

Centralized Workflow using a central or main repository in which all changes are committed and pushed to. No other branches are needed like in Gitflow Workflow. This workflow tends to be better suited for smaller teams as there is no complex pull request or forking patterns that are set in place that you would need for larger teams. The way that a Centralized workflow would begin is by having each individual developer clone the central repository otherwise known as the main branch. A repository is a container that is used to hold the files that are being edited and created by the developers.

After a developer clones the central repository they will have a local version that they would be able to make and commit changes to without worrying about other developers being affected. A commit also presents the developer with the chance to leave comments on what exactly they changed within the files. These commits are not sent to the central repository until said developer decides to push those changes. Pushing a commit updates the central repository and therefore shares it with the other developers working on the project. However, something to keep in mind when pushing a commit, is that it could potentially contain a conflict with someone else's code within the central repository.

In a situation in which a conflict is present, it means that the pushed commit would overwrite changes made by already official commits. This would result in git refusing to push this developers commit to the central repository. What would need to be done in this situation is a “git pull”. This is where a developer pulls the most recent version of the central repository. In simple terms it is a developer saying they would like to add their work to what their fellow developers have already done. This ensures that the history remains linear which would help prevent any conflicts within the official code in the first place. This process is also called rebasing which would assist in catching these merge conflicts with each commit.

However, a conflict could still be present after rebasing if the code being pushed by two developers are working on a similar feature if not the same feature. What this would result in is something called a merge conflict. To solve this one of the developers could run “git status” which would show them where the problem is located. This developer could then edit the files and resolve the file until they are happy with the result. After that they would use a command called “git rebase” and recommit the files. This would handle the process of rebasing the code and then continue looking for conflicts. Once the conflicts are all sorted the developer would simply just need to push the code for it to be published to the central repository.