Git Hub Workflow Essay by Jake Cubernot

Git - A modern version control system that is by far the most used of its kind worldwide by both independent developers' projects and commercial projects. Git was developed by Linus Torvalds in 2005, who was also the creator of the well-known Linux operating system, kernel. Modern version control systems are used to help developers collaborate on projects by being able to move work into a remote repository. However, Git saves each time a developer moves work to the remote repository, creating the ability for developers to revert to older versions of their work in case of any conflict. Without Git and other modern version control systems, developers would have a much higher risk of their projects being erased due to unintentional reasons and it would make it much more difficult for multiple developers to work together.

Git Workflow - The Git Workflow is the process by which how a project gets moved around the Git repository and individuals' personal computers. The Git Workflow can be well-visually represented by different kinds of diagrams. The workflow shows the different actions an individual can use to move their project to specific parts of the Git, this includes the working directory, staging area, local repository, and central repository. The working directory is usually the IDE in which the user is working on their project. The staging area is the location where any changes or new code are sent and can be reviewed before adding to the repository. The staging area will display errors between the user's new work and their local Git's work like if there are any merging errors or work that may get overwritten by the new work. Once the work is cleared from the staging area, it is added to the local repository which is the final stage before moving to the central repository. The local repository will also give similar errors if any issues occur between the local and central repository. If approved, work will be sent to the central repository in which all of the project's users can view, edit, or share the new work added to the Git repository.

Commits - Commits are the process of moving work from the staging area to the local repository. During the commit process, errors can occur if specific conditions are not met. The commit would then send the work to the final stage before moving it to the central repository.

Pushes - Pushes are the process of moving work from the local repository to the central repository. After work is committed from the staging area and is validated for the local repository, the work will then be moved by the push process. If the work is validated, it will proceed to the central repository.

Pulls - Pulls are the process of moving work from the central repository to the working directory. Pulls are incredibly useful as they skip a couple of the stages so the user can quickly work on the central repository in one process. Without pulls, a user would have to deal with a multi-process action to begin their work which may slow down the ultimate production of a project.

Merges - Merges are the process between any stages in which work is being stored and ensure that new work can be combined with older work within a project. Conflicts may arise during merges to help make sure nothing within the project gets deleted or any unwanted work gets added.

Conflicts - Conflicts usually occur during the merging process of the Git called merge conflicts. One general example of a merge conflict is when two different users try to

merge work into the same section of a project in the central repository. This could result in the overriding of one's work, which would be deleted, but because of Git, preventative actions are taken to prevent these kinds of issues.

Repositories - A repository is essentially the central storage location for managing data and files. A repository can be located either on a user's local device or on the cloud with Git. Unlike regular storage systems like a folder on a local device or a folder on a cloud drive, repositories will keep track of what is being sent in and out of it and manage any conflicts within the repository.