

Github

Git is the most popular modern version control system in the world currently. A large number of software projects rely on Git for several different reasons. These include version control, and the ability to share projects easily. Git workflow is vital to ensure fast delivery among teams. Many developers are in development teams and need these workflows to be more organized and effective. A centralized Git workflow allows all developers to make changes directly to the main branch, and this includes a history log of every single change made. This is especially effective for any group work, but developers must keep unstable changes local until they are finished working. Each Git workflow has different advantages, including enhancing the quality of the code base and stabilizing the production application.

Some common terms in Github include commits, pushes, pulls, and merges. The git push command updates the remote branch with local commits. A common synonym for git push is update or publish. It only updates the corresponding branch on the remote, so if one is on the main branch when they execute git push, then only this main branch will be updated. A good practice to avoid a mistake here is to use git status to see what branch is being worked on before pushing to the remote. Another important thing to note here is to run the git pull command before you push any new changes to the remote. Doing so will update the local branch with any new changes that may have been pushed to the remote by anyone else. The git pull is among the four remote operations within Git. This operation is important because without it, your local repository will never be updated with changes from the remote. This is why the pull operator is amongst the most used Git commands. Moving forward, the Git merge command permits one to take independent lines of development produced by the Git branch and combine them into a lone branch. Git merge works by combining multiple sequences of commits into one single history. It is most often used to combine two separate branches. For example, if one has a new branch feature that is based off the main branch and wants to merge this feature branch into main, they will need to invoke this command to do so. This command comes with some issues, resulting in Git merge conflicts arising. These conflicts arise when two people have changed the same lines in a file, or if one deleted a file while someone else was modifying it. In these cases, Git cannot automatically determine what is correct. These conflicts will affect the developer conducting the merge while leaving the rest of the team unaware. As a result, Git will mark the file as being conflicted and halt the merging process leaving the developer to resolve this issue. To identify a merge conflict, one can run the git status command so Git can produce some output letting the user know a conflict has occurred.

Lastly, a Git repository is a virtual storage of one's project. It allows one to save versions of their code, and the ability to access this code whenever they choose to. To create a new repository, one can use the git init command. This is a one-time command during the initial setup of a new repository. It will also create a new main branch. Users are also allowed to clone

existing repositories with git clone. This is done simply by using the git clone <repo url> to create a copy or clone of the repository. To save changes to this new repository, one can use git add and git commit to do so. This will allow the user to change their files and projects very effectively. The user is also allowed to see the history and track updates to the file being made.

References: <https://www.atlassian.com/git/tutorials/what-is-git>