Git workflow is basically the standards and methods of operation a team uses for their repository when working on a project. It’s a way to keep organized and promote a less chaotic environment. With that being said, there are multiple different versions of Git workflow, and each has its pros and cons. In this report, pushes, pulls, merge conflicts, branching, and other concepts will be covered.

First, when creating a repository you’ll need to consider your branches. Branches are essentially different versions of your repository that can each be edited without affecting each other. By default, you have one master branch that’s essentially the final and most important one. The other branches are essentially copies of the master branch that get edited and generally need approval before they can ever merge into the master branch. Merging is essentially bringing the changes over from one branch to another. Ideally, the master branch would receive very stable changes and would improve from a merge with another branch. Without the use of branching for creating new features or fixing bugs, one may end up changing up the master branch for the worse and harm the project for everyone involved.

Now how do you make changes to a repository in the first place? First, you can clone the repository so you have your own local copy to work on. It essentially creates an entirely new copy of the repository that you’re free to edit and change how you want. You’ll want to start making your changes, and once you’ve made enough you can try and commit. Committing basically records the changes you’ve made to the repository, and it usually comes with some comment or description to explain what the commit is actually doing. Perhaps someone was adding a new class or was simply adding JavaDocs to a few files. Once your changes are ready, you should try and pull the GitHub repository. Pulling the repository means that you’re merging any changes anyone has made to the repository to your local repository. This may result in merge conflicts which are basically caused when GitHub can’t figure out how to properly work the changes into your repository. You’ll have to resolve these yourself, but afterwards you’re able to push your changes to your branch.

However, in order to merge a branch with someone else’s branch, a pull request must be opened. Generally in a company, your superiors will be overseeing pull requests and checking over code to ensure that your changes won’t be detrimental and are actually good. It’s entirely possible for your pull request to be rejected, though hopefully you’ll receive some guidance and comments on why. However, if you happen to be the one controlling the project then you can simply confirm your own merges.

Essentially, Git workflow is a tried and tested concept that will ultimately improve communication and collaboration between teams. If everyone is on the same page, it leads to much more organized and efficient work than if people were to do things on an ad hoc basis.

Works Cited

“Using Github to Share with SparkFun.” *Using GitHub to Share with SparkFun - SparkFun Learn*, https://learn.sparkfun.com/tutorials/using-github-to-share-with-sparkfun/all.

User, Devmountain. “What Is Github and How Do You Use It?” *Devmountain*, 8 Aug. 2022, https://devmountain.com/blog/what-is-github-and-how-do-you-use-it/.