Matthew Ernst

Git Workflow

‘Git Workflow’ is a standard and essential practice used by software developers and any other sort of programmer whenever multiple entities or people are editing and modifying code. This approach provides an easy to understand and structured approach to the overview and administration over a code space.

It starts with the software developers ‘cloning’ the central repository on their own machine, which creates an environment on the user’s device where you can modify, debug, and improve code. Once the code is ready to be uploaded to the actual origin repository, the developer is given the option to ‘push’ it to a branch, or the main branch. Since it isn’t the best idea to let untested code get put online, you can create a new branch centralized around that feature until it is done and safe to be pushed into the real world and then eventually push it over to the main branch.

When the developer is ready to push code to the central repository, they will create a pull request. After a pull request is created, other developers will then have to approve your changes. Creating a pull request also creates an in-depth overview of any changes that your code introduces compared to the original repository, which makes it easy to catch any unintended changes that you almost let see the light of day.   
 Once the pull request has been approved by code reviewers, the developer is ready to proceed and push or merge the code to the central repository. At this point the developer would have to be mindful and be on the lookout for any conflicts that appear. These conflicts can appear when any merged code contradicts each other. If these conflicts are made to the live build, the conflicts can cause discrepancies between what the code is expected to do and how it actually performs.