**Fonk Assessment**

Git Workflow

Central Git Repository

The first and foremost action the lead developer should take on the beginning of a new development project is creating the project structures, which includes the setup and structure of the Git repository. Doing this at the start of the project allows each developer on the project to clone a local copy of the repository onto their machine so they can contribute to the project.

Git Workflow Guidelines

As a team, some rules should be applied with regards to the repository.

Examples:

1. Before your day coding day begins, do a pull from the Main repository to get all latest changes and commits
2. Make sure you do a commit at the end of your day to make sure the latest code always goes into the repository.
3. If a component or feature is not complete, create a branch for your feature so your code can be committed separately from the main branch.
4. Once your feature/component is completed, merge master into your branch and resolve any conflicts, the branch should then be fully tested for the component / feature you have built, then merge your branch into master. This assures that there will be no conflicts once your code is implemented into the master branch.
5. Remove any unused branches once you are done with a branch to avoid a buildup of unused branches.
6. Any conflicts should be resolved correctly and manually to oversee that no code gets overwritten.

Code Reviews:

In an ideal Git Workflow, code reviews should be implemented. This would mostly affect a larger project with many developers on the project to make sure that the code committed into the repository is free from bugs or code that might be able to be streamlined more efficiently. With a code review, developers on your team could comment any foreseeable code discrepancies or perhaps a more efficient way of doing the component / feature.

This also alleviates any bugs that may occur in the code.

Summary

Having your team understand your Git Workflow ensures that they adhere to the guidelines so that there is less work involved in maintenance of the git repository and less chance that anything major might happen to disrupt that workflow.