

# Development Repository Structure and Workflow

We are thinking that we will have 2 long lived branches (test and master) and potentially 1 optional medium-lived branch called "release" and then there will be n short-lived branches for n features and hotfixes.

The feature branch should be created from the test branch by default. If a feature needs to be expedited before existing features, then you should probably create the feature (or hotfix) branch from master.

After the developer pushes code into the feature branch the CI process is triggered. Once building, deployment into test environment, and testing are completed, then the developer creates a pull request for peer review. Once peer review is completed, the developer merges their changes into the test branch.

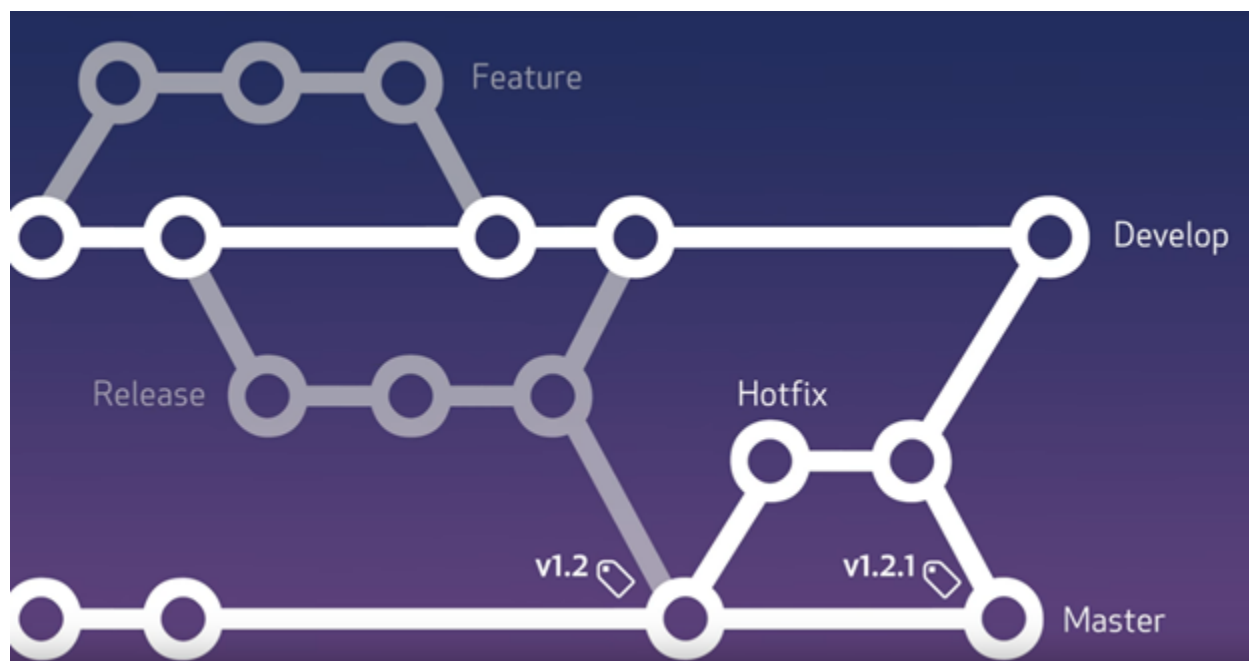
Things to discuss:

Multiple repositories or a single repository?

Where does selenium testing come in?

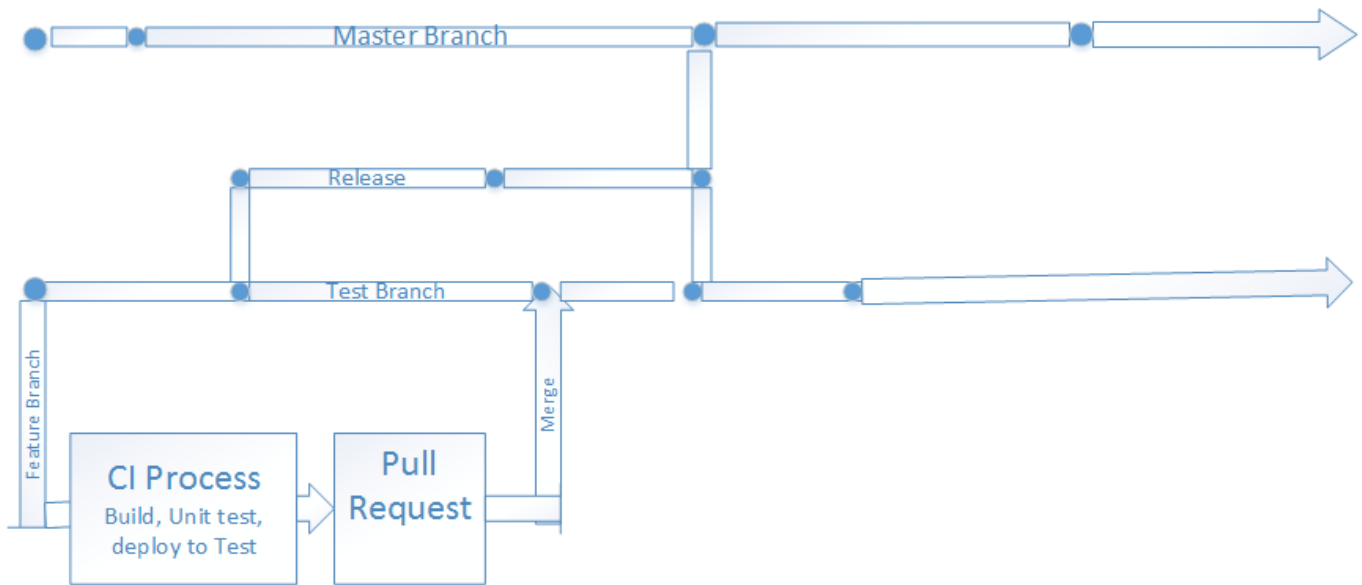
Resources

<https://resources.github.com/videos/github-best-practices/>  
<https://www.youtube.com/watch?v=aJnFGMclhU8>

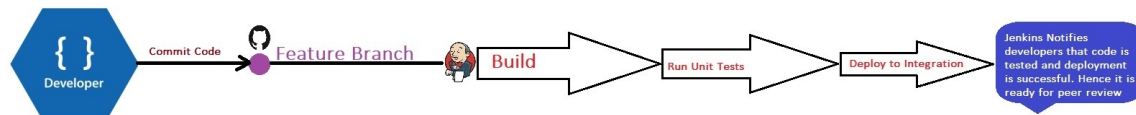


## Action Item

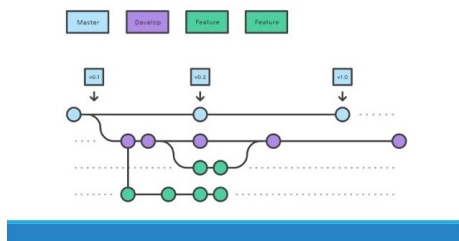
- Developers are going to do research on single repository vs multiple repository
- Siva is going to provide Diagram of basic git work flow
- Siva is going to do research on Selenium testing work flow and where does it fit in



**Pull Request\*** : Developer creates the pull request for code review and UI testing. UI tester will write test cases after code review passed and he will approve the pull request and merge into test branch.



## Feature Branches



## Release Tags

