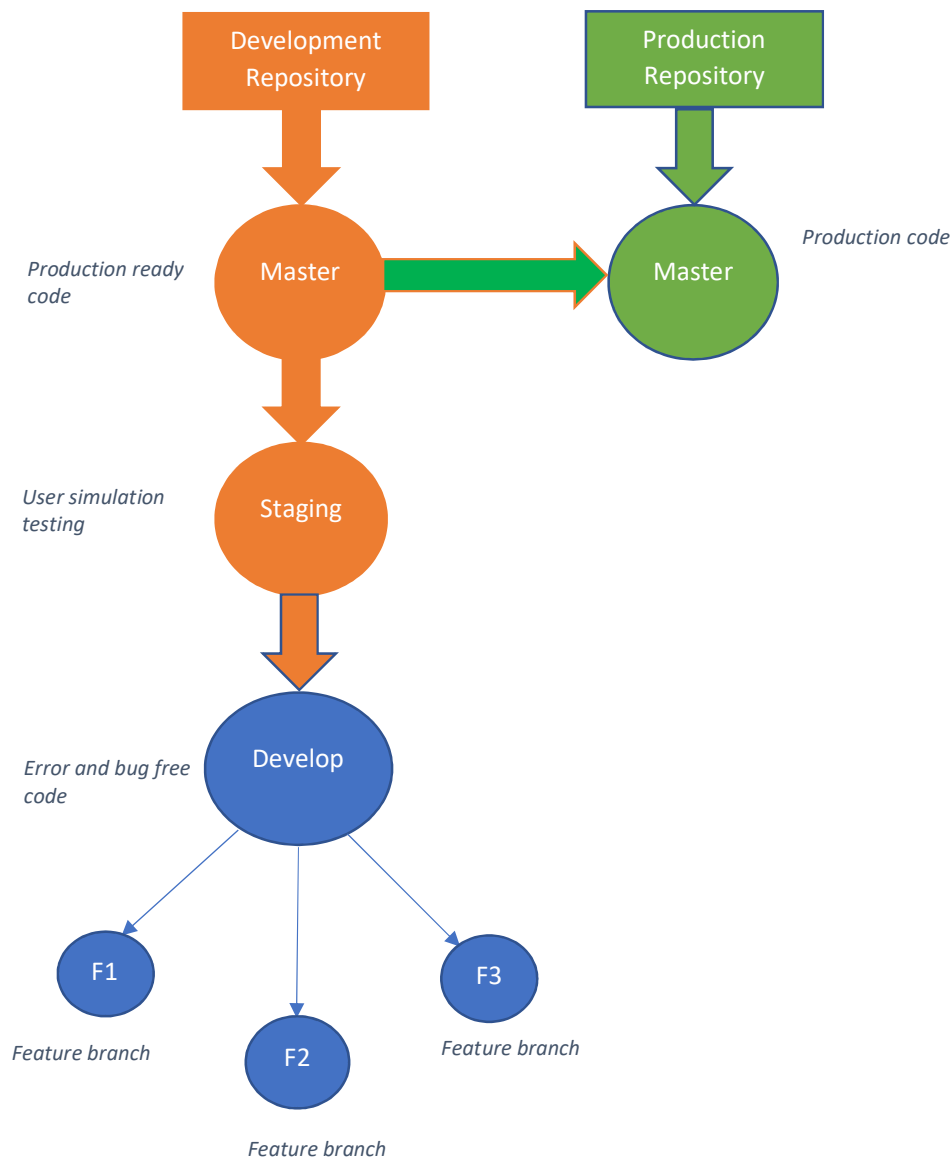


Git workflow & Development process

AWS CodeCommit is a version control service hosted by Amazon Web Services that you can use to privately store and manage assets (such as documents, source code, and binary files) in the cloud. [Learn more](https://docs.aws.amazon.com/codecommit/latest/userguide/welcome.html) [https://docs.aws.amazon.com/codecommit/latest/userguide/welcome.html].

Git workflow

A Git Workflow is a method of using git to accomplish work in a consistent and productive manner. Whenever multiple developers are involved in a project it is necessary to use the right workflow for git. [Learn more](https://www.atlassian.com/git/tutorials/comparing-workflows) [https://www.atlassian.com/git/tutorials/comparing-workflows].



Branches in development repository

Master:

- ✓ Master branch should always have a copy of production ready code.
- ✓ No one including the tech lead should be directly coding in master branch
- ✓ Only team lead will have access to master branch.

Staging:

- ✓ Staging branch is created from the master branch and it should always have a copy of release ready code.
- ✓ No one including the tech lead should be directly coding in staging branch
- ✓ Only team lead will have access to staging branch
- ✓ Features integration should be properly tested before merging into staging branch

Develop:

- ✓ Develop branch is created from the staging branch and it should always have a copy of tested and error free code.
- ✓ Everyone in a team will have access to develop branch
- ✓ No one including the team lead should be directly coding in develop branch
- ✓ Feature should be properly tested before merging into develop branch

Feature:

- ✓ Feature branch is created from the develop branch
- ✓ Everyone in a team will have access to feature branch
- ✓ For every feature that is built in the application including testing, bug fixes or other reasons a separate **feature branch** is created. This ensures that the features can be built independently
- ✓ All the developer code in their individual feature branches

Branches in production repository

Master:

- ✓ Master branch should always have a production code.
- ✓ Only team lead will have access to master branch.

Git Tips

How to create a branch?

- ✓ Create your feature branch always from **develop branch**.
- ✓ Command – `git checkout feature`

How to merge a branch?

- ✓ Go to **develop branch**
- ✓ Pull all the changes in develop branch [`git pull`]
- ✓ Merge your feature branch [`git merge feature`]
- ✓ Resolve merge conflicts if any
- ✓ Commit and push all your changes

What we should **not** do?

- ✓ Commit into others feature branch
- ✓ Merge into others branch
- ✓ Commit without resolving merge conflicts
- ✓ Git pull from branches other than develop branch

Helpful Links

- ✓ [Useful Git commands](https://github.com/bpassos/git-commands) [https://github.com/bpassos/git-commands]
- ✓ [AWS CodeCommit](https://docs.aws.amazon.com/codecommit/latest/userguide/getting-started-cc.html) [https://docs.aws.amazon.com/codecommit/latest/userguide/getting-started-cc.html]

Authored by:

Ravishankar S R
rr@zaubacorp.com