**Git Standards**

1. Branch Naming

Use meaningful names for branches. There should always be a master(stable) branch and a working branch where the development take place, and all feature branches should be created off the develop branch.

1. Commit Messages

Write clear and concise commit messages. Use the imperative mood in commit messages (e.g., "Fix a bug" instead of "Fixed a bug"). This will help teammates follow along with the development.

1. Code Review

Perform code reviews for all changes. Ensure that the code adheres to the coding standards. This also helps that the code is correctly formatted and merged into the correct branches. This keeps the code up to date and without errors.

1. Version Tags

Use version tags to mark significant release points (e.g., "v1.0.0"). This keeps track of the best stable version and releases.

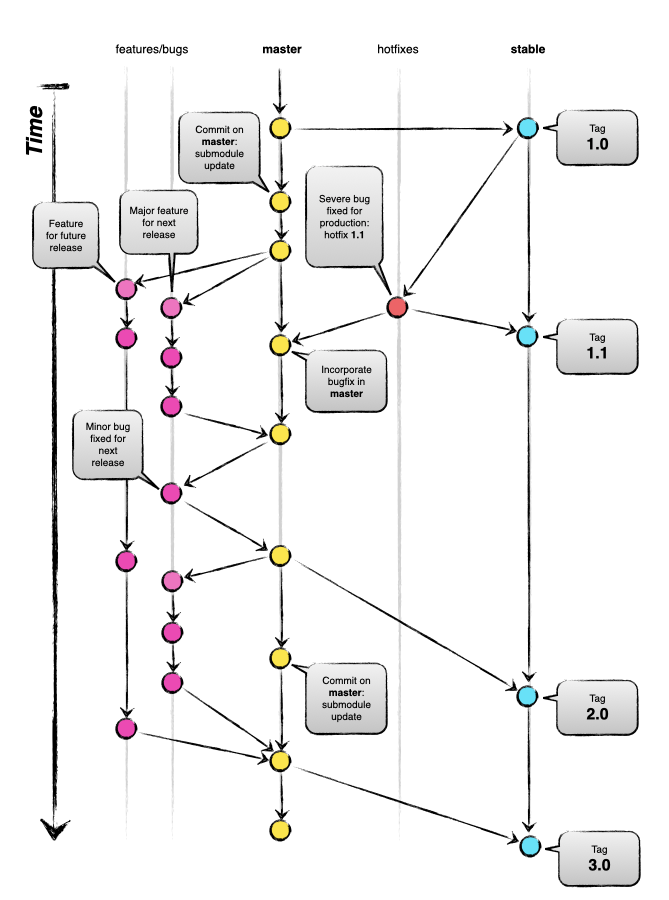
1. Git Workflow

Follow a Git workflow (e.g., Gitflow) to structure your repository and manage feature development, releases, and hotfixes.

1. .gitignore

Create a .gitignore file to specify files and directories that should not be tracked by Git.

**Workflow Diagram**

[](https://raw.githubusercontent.com/digitaljhelms/public/master/gitflow-model.png)