

Metro Bus – CS203 CPD Project Development

Metro Bus Web Application Development

Table of Contents

[Project Rules and Conventions 3](#_Toc137201082)

[Git Workflow 3](#_Toc137201083)

[Branching Strategy – Git Flow 3](#_Toc137201084)

[Git Flow Naming Conventions 3](#_Toc137201085)

[Commit and Pull Request Rules 4](#_Toc137201086)

# Project Rules and Conventions

## Git Workflow

Git is a distributed version control system that is commonly used in software development projects because it allows developers to collaborate on projects with its massive arrays of tools in a safe and organized manner. For Metro Bus Application development, we will be using git to allow us to collaborate in developing our project with the most up to date code and documentation . Benefits of using Git includes…

* Git allows you to see the changes made to the code and undo them if needed.
* Git lets multiple people work on the same code at the same time.
* Git is a safe and easy way to manage changes made to the code.
* Git has built-in tools that make it easy to create different versions of the code and merge them back together.

### Branching Strategy – Git Flow

Git Flow is a git Branching model that involves the use of feature and branches and multiple primary branches [(Driessen, 2010 - 2020)](https://myacg-my.sharepoint.com/personal/270168718_yoobeestudent_ac_nz/Documents/202CPD_Documentation.docx#_Reference_List).

Diagram

Description automatically generated

How it works is instead of pushing every development into the main branch to record the history of the project. A development branch would be created where a specific development will be branched and worked upon so that multiple developments can be done at the same time in a more focused manner, where developers push features and fixes onto the development branch instead of the main. Steps to do these are…

1. git branch develop – create development branch
2. git push -u origin develop – push development branch into remote repository

### Git Flow Naming Conventions

In summary the hierarchy of git flow workflow development branching is…

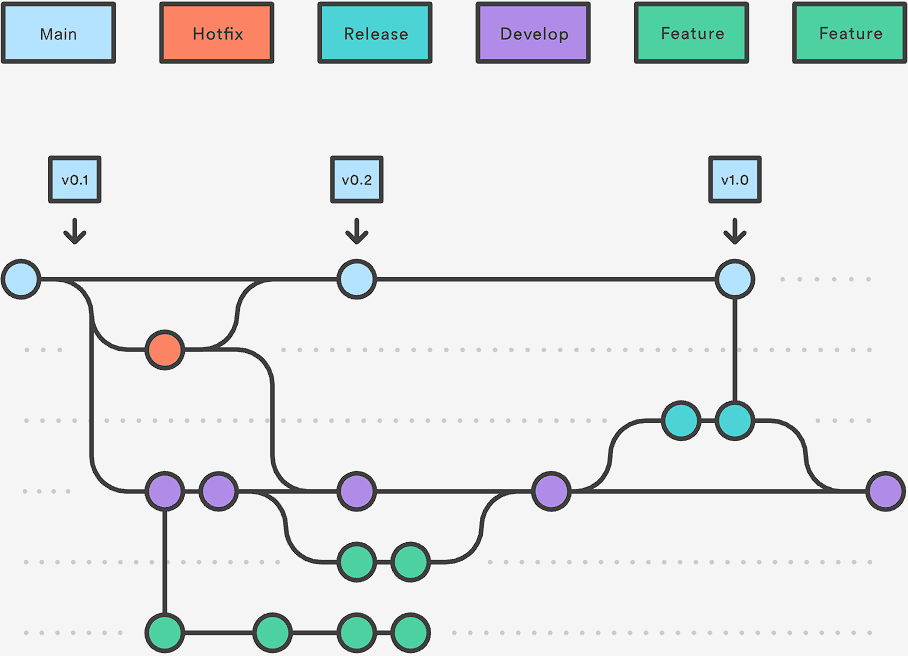
Main Branch -> Development Branch -> Feature Branches -> Hotfix Branches & Support Branches

When releasing a version, a branch cloned through parts of main will be used to create a release branch which will then receive Hotfix & Support Branches.

When naming a branch, it must have the type of branch it is and its purpose. E.g., Feature/Create Rules…

* Development Branches? [development/purpose]
* Feature Branches? [feature/purpose]
* Release Branches? [release/purpose]
* Hotfix Branches? [hotfix/purpose]
* Support Branches? [support/purpose]

Example:



The overall flow of Gitlow is….

1. Create a develop branch from the main branch.
2. Create a release branch from the develop branch.
3. Create feature branches from the develop branch for new features or changes.
4. When a feature is complete, merge it into the develop branch.
5. When the release is complete, merge it into both the develop and main branches.
6. If an issue is detected in the main branch, create a hotfix branch from main.
7. Once the hotfix is done, merge it into both the develop and main branches.

### Commit and Pull Request Rules

Creating a commit and a pull request to upload work into the main repository is a very important part of git development and must be done right to keep track of work development easily. To do so commit / pull name and message must done properly so work done can be understood properly and efficiently without confusion.

To do so the title of the commit / pull must first start with a verb (added, deleted, changed) followed with what was done / purpose of the commit / pull. The message must also contain why the commit / pull was done followed by a list of the changes that was done. The list should also contain lists of reasons of things to look out for in the pull / commit.

Example…

|  |  |
| --- | --- |
| Title | Added documentation on initial concepts. |
| Message | This was done to understood what needs to be worked upon in the future.   * Added purpose, functions and worth of the project continuation. * Some images are lagging out. |