Git Repository Standards

# Git branch naming conventions

## Use issue tracker IDs in branch names

e.g. 722-add-billing-modul, 723-fix-highlighting

## Add a short descriptor of the task

* Use a short, actionable descriptor of the task after the issue ID. This makes the branch name recognizable, distinct, and easy to search for in case you don’t have the issue ID handy.
* Make sure that the descriptor is concise, but descriptive enough to give you an idea of what’s going on in the branch.

## Use hyphens as separators

* This is a little opinionated, but hyphens make for good separators in branch names. You could use an underscore, \_, too.
* The key is to be consistent, though.

## Start branch name with a Group word

Have a look at the below examples:

1. **bug-logo-alignment-issue** – the developer is trying to fix the logo alignment issue;
2. **wip-ioc-container-added** – the branch relates to the task to add an IoC container in progress.

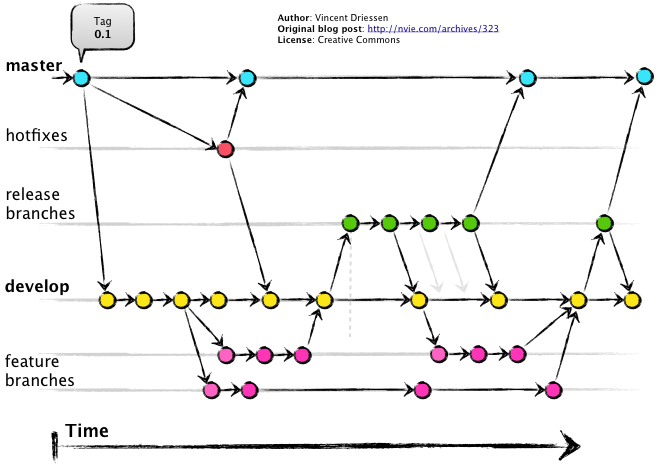
WIP – Work in Progress

# Branch types

There are several types of branches that are frequently used in software development.

|  |  |  |
| --- | --- | --- |
|  | **Development branch**  Usually the integration branch for feature work and is often the default branch or a named branch. For pull request workflows, the branch where new feature branches are targeted. | master  or  develop |
|  | **Production branch**  Used for deploying a release. Branches from, and merges back into the development branch. In a Gitflow-based workflow it is used to prepare for a new production release. | varies |
|  | **Feature branch**  Used for specific feature work or improvements. Generally branches from, and merges back into, the development branch, using pull requests. | feature/ |
|  | **Release branch**  Used for release tasks and long-term maintenance versions. They are branched from the development branch and then merged into the production branch. | release/ |
|  | **Bugfix branch**  Typically used to fix Release branches. | bugfix/ |
|  | **Hotfix branch**  Used to quickly fix a Production branch without interrupting changes in the development branch. In a Gitflow-based workflow, changes are usually merged into the production and development branches. | hotfix/ |

# Gitflow



Gitflow is a very popular workflow that defines the following types of branches:

* **master**: contains production code.
* **develop**: contains the latest development changes that will be included in the next release.
* **feature**: a new branch is created for each new feature we work on. We start it from develop and once we’re done, merge back into it.
* **release**: starts from develop and signifies that there will be a new release once we merge this branch into master.
* **hotfix**: used when we need to deliver urgent changes to our production app but develop is not yet ready to produce a release branch. Starts from master and merges into both master and develop.

<https://proandroiddev.com/how-to-set-up-an-efficient-development-workflow-with-git-and-ci-cd-5e8916f6bece>

