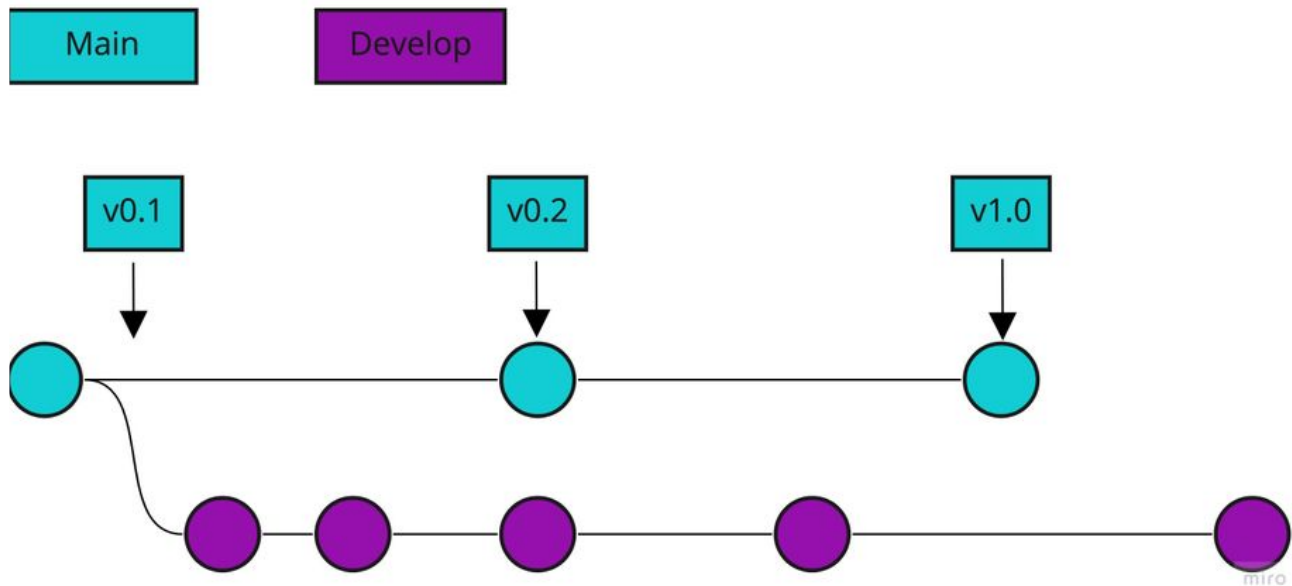


Git Workflow

How it works



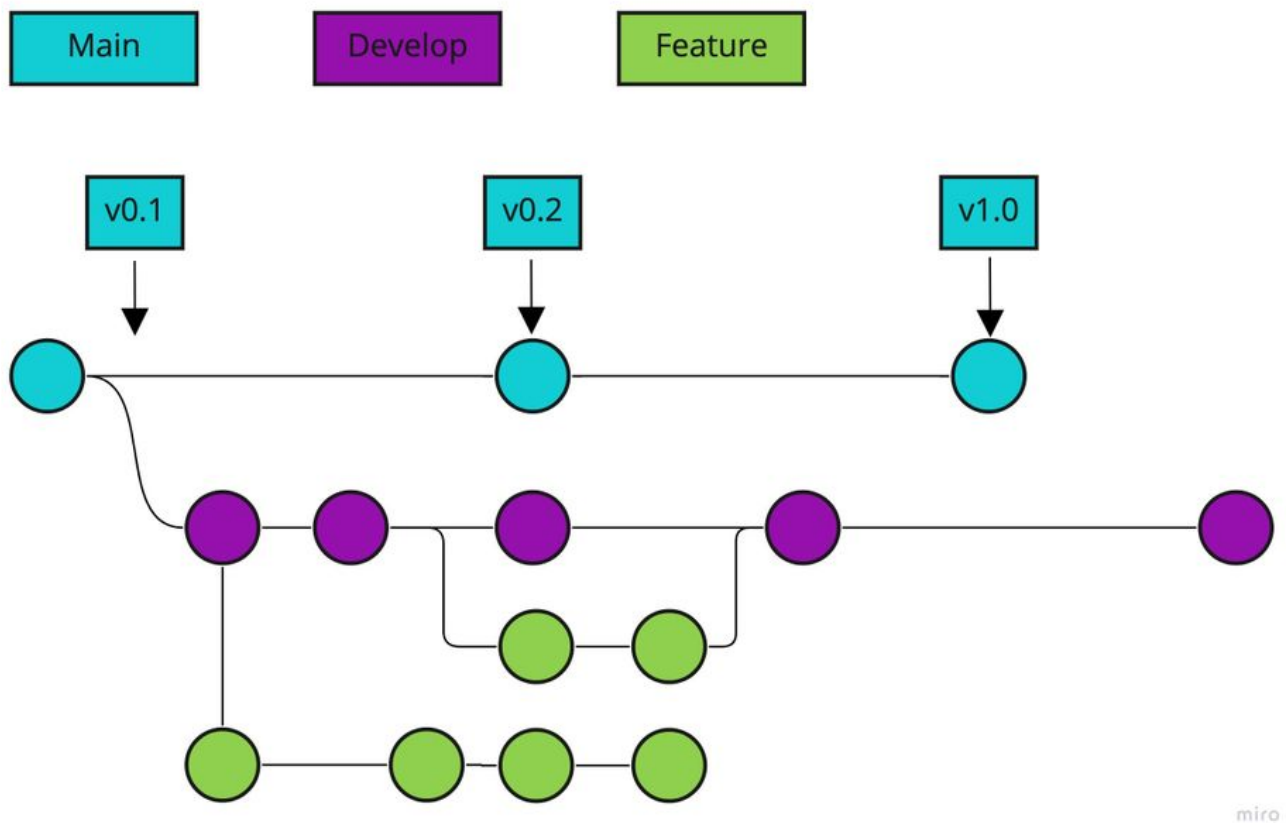
Develop and main branches

Instead of a single `master` branch, this workflow uses two branches to record the history of the project. The `master` branch stores the official release history, and the `develop` branch serves as an integration branch for features. It's also convenient to tag all commits in the `master` branch with a version number.

The `develop` branch will contain the complete history of the project, whereas `master` will contain an abridged version. Other developers should now clone the central repository and create a tracking branch for `develop`.

Feature branches

Each new feature should reside in its own branch, which can be pushed to the central repository for backup/collaboration. But, instead of branching off of `master`, feature branches use `develop` as their parent branch. When a feature is complete, it gets merged back into `develop`. Features should never interact directly with `master`.



miro

Note that `feature` branches combined with `develop` branch is, for all intents and purposes.

`Feature` branches are generally created off to the latest `develop` branch.

Finishing a feature branch

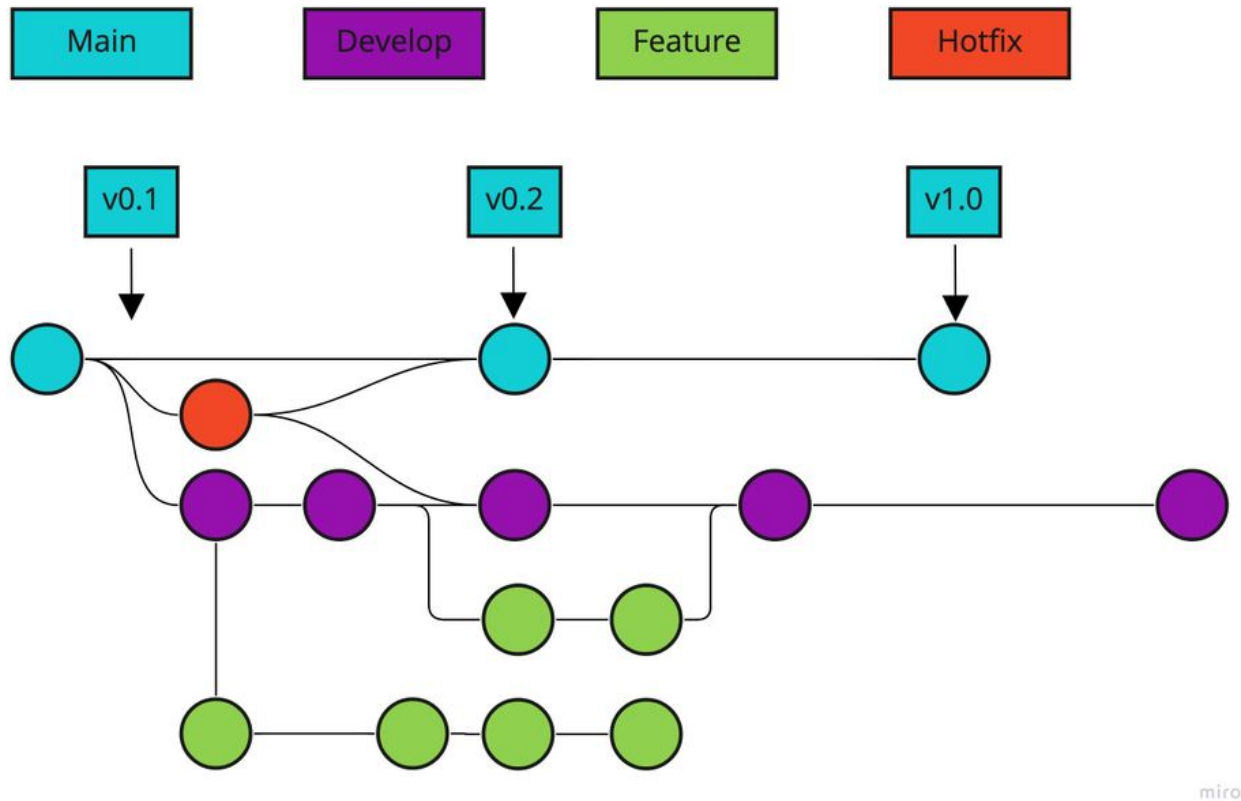
When you're done with the development work on the feature, the next step is to merge the `feature_branch` into `develop`.

This will happen through pull request.

Release

Once `develop` has acquired enough features for a release (or a predetermined release date is approaching), we should open pull request from `develop` to `master`.

Hotfix branches



Maintenance or “hotfix” branches are used to quickly patch production releases. Hotfix branches are a lot like `release` branches (which we do not use) except they’re based on `master` instead of `develop`. This is the only branch that should fork directly off of `master`. As soon as the fix is complete, it should be merged into both `master` and `develop`, and `master` should be tagged with an updated version number.

The overall flow of Git is:

1. A `develop` branch is created from `master`
2. `Feature` branches are created from `develop`
3. When the `feature` is complete it is merged into the `develop` branch
4. If an issue in `master` is detected a `hotfix` branch is created from `master`
5. Once the `hotfix` is complete it is merged to both `develop` and `master`