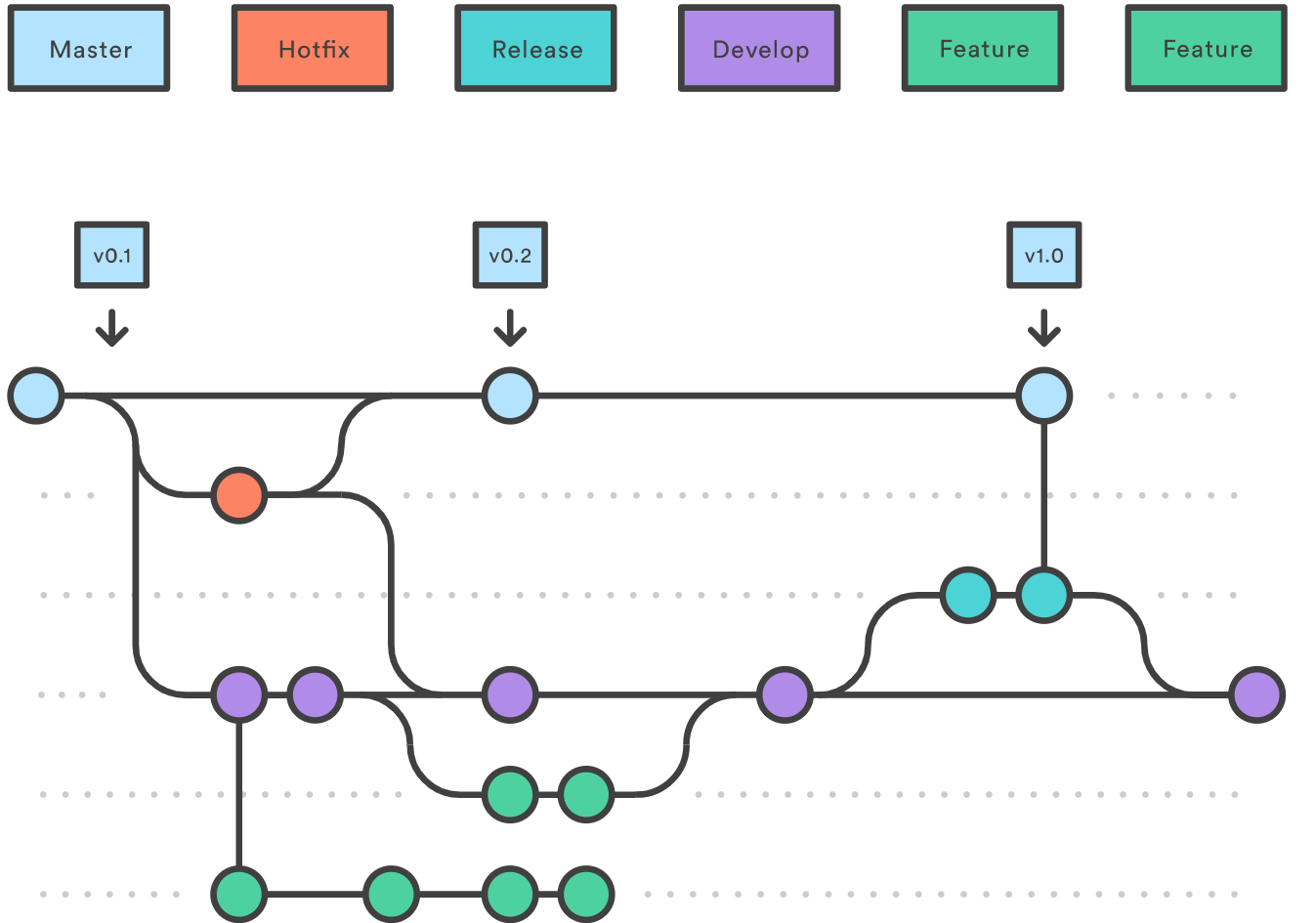
tessagon branching strategy  
GitFlow workflow branching

short line

# Branch Hierarchy:

1. master
   1. hotfix
2. release
3. develop
4. feature

# Motives:

1. Use **feature** branches for all new features (functionalities) and bug fixes.
2. Merge feature branches into the **develop** branches using pull requests.
3. We merge all develop branches into a **release** branch before the end of each sprint.
4. Keep a high quality, up-to-date master branch **after every release**.

# Feature:

* **Parent branch**: Develop
* Each new feature resides in its own branch
* When a feature is complete, it gets merged back into develop.

# Develop:

* **Parent branch:** Master
* Serves as in integration point for all features
* Once all features are complete, develop is merged to release.

# Release:

* **Parent Branch:** Develop
* Records history of every sprint cycle:
  + 3 release branches for each sprint
* No new features should be implemented in a release branch
  + Only major bug fixes.
* Once release is ready (bugs and other small fixes), we merge to master
  + This should coincide with the end of a sprint
  + Good practise to have version number in each commit - so that it’ll be in master

# Master:

* **Root branch**
* Records the official release history
* Only updated after each sprint - this will be what the client sees

# Hotfix:

* **Parent branch:** Master
* Urgent bug fixes or changes made out normal developmental process - **risky**
* Once complete it is merged to both develop and master

# Overall workflow:

1. When a feature is complete it is merged into the develop branch
2. When the release branch is done it is merged into develop and master
3. If an issue in master is detected a hotfix branch is created from master
4. Once the hotfix is complete it is merged to both develop and master

# Naming Convention:

Consistent naming convention for your feature branches to identify the work done in the branch. You can also include other information in the branch name, such as who created the branch. Some suggestions for naming your feature branches:

* users/username/description
* users/username/workitem
* bugfix/description features/feature-name
* features/feature-area/feature-name hotfix/description
  + user/login/google-authentication/third-party-login-successful
* release/version/description
  + e.g. release/1.4.0/login-fixed

# Moving forward:

* Master (don’t need to use yet)
* Release (don’t need to use yet)
* Develop
  + Jono + backend-testing
* Feature (all seperate branches) e.g.
  + login/sign-up functionality (backend)
  + Google-authentication (backend)
  + profile template (frontend)
  + login/sign-up forms (frontend)

# More info:

* [Gitflow Workflow](https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow)
* [Git branching guidance - Azure Repos](https://docs.microsoft.com/en-us/azure/devops/repos/git/git-branching-guidance?view=azure-devops)
* [Git Flow is superflous and complex](https://www.franzoni.eu/git-flow-is-superflous-and-complex/)
* [Professional Guides: Workflow Strategies](https://www.youtube.com/watch?v=aJnFGMclhU8)