<Coding Club>

GITFLOW WORKFLOW

BY VINCENT DRIESSEN

IT'S JUST GIT

BUT THERE IS A COMMAND LINE UTILITY

Speaker notes

Caveat, I almost always work alone. I may not be aware of common issue using this strategy in a group. Also, this isn't a solution for every occasion. I often use a single branch.

WHY?

- Prevents orphaned files and branches
- Allows for easy, confident work on
  - multiple features
  - hot fixes
  - deployment to the real world
- It can be reasoned about

Speaker notes

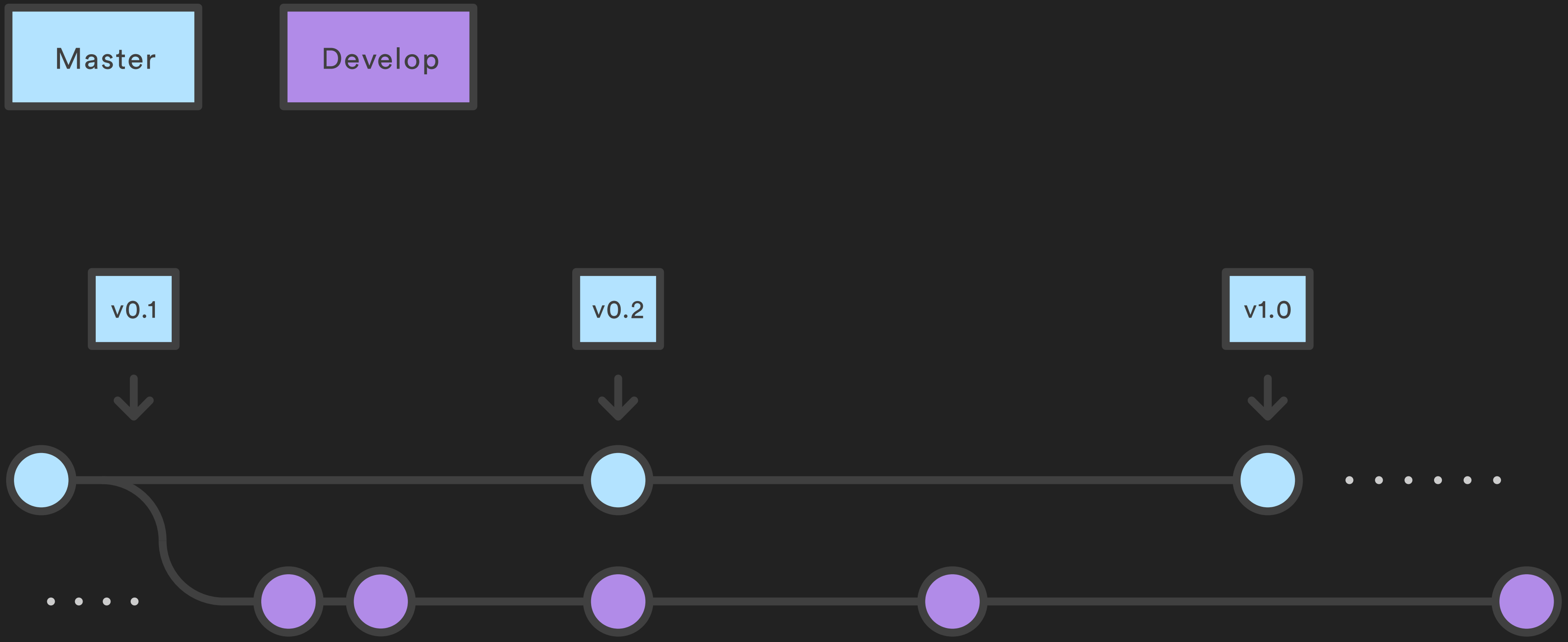
Even when using versioning it can be easy to get lost. I often mindlessly make commits without thinking about the larger structure. You want to think about structure at breaking points. Working on multiple features at a time can lead to having orphaned files and unnecessary code comments to keep track of what you are doing.

DEVELOPMENT BRANCH

Development branch is a comprehensive history of the project.

Speaker notes

First thing we need to do is complement our master branch with a development branch. The names of the branches can be whatever makes most sense to you but this is the community convention.


Master
Develop
v0.1
v0.2
v1.0

MASTER BRANCH

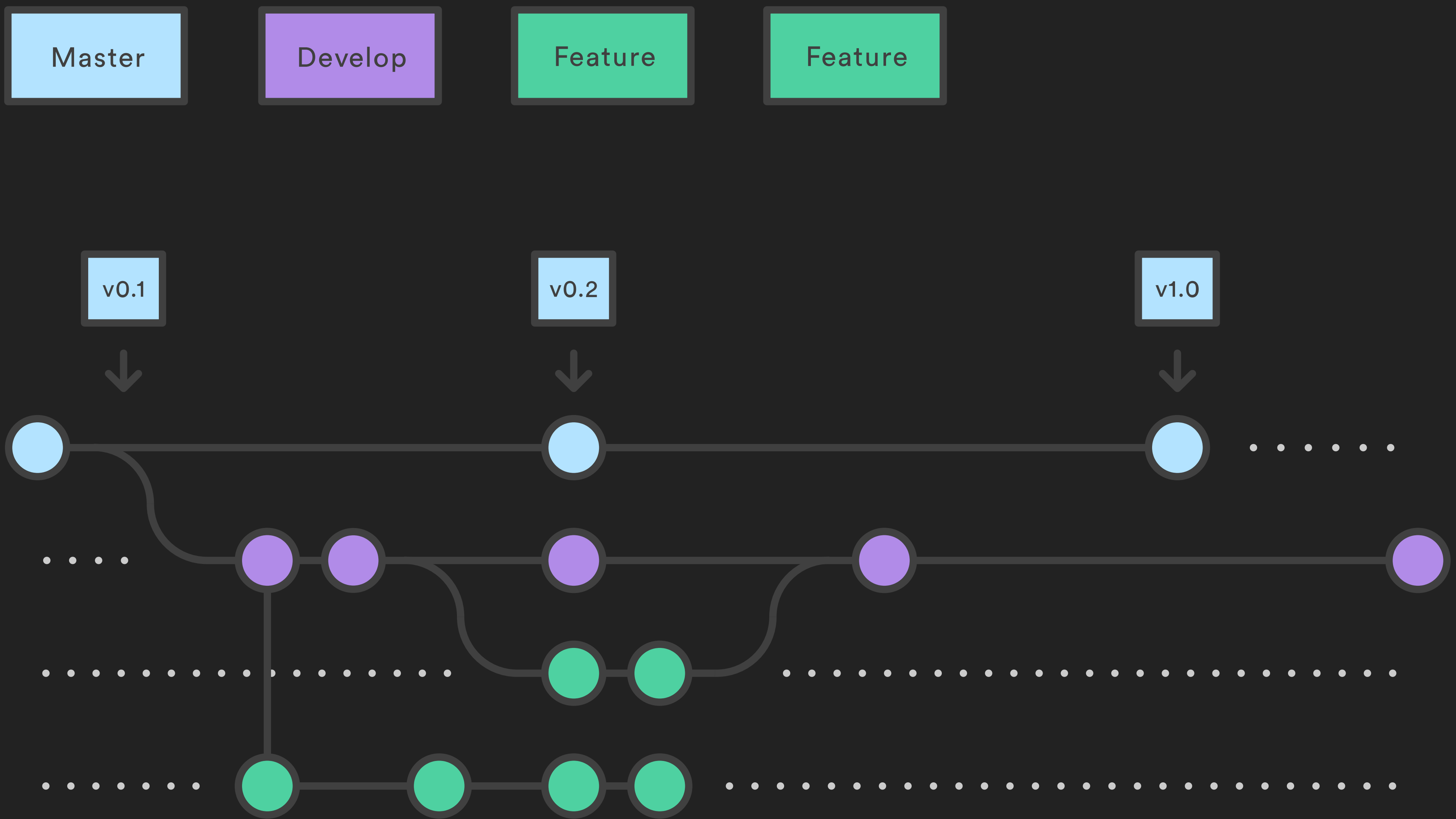
The Master branch will hold all the public releases of the project.

FEATURE BRANCHES

Feature branches start and end on the Development branch.

Speaker notes

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.

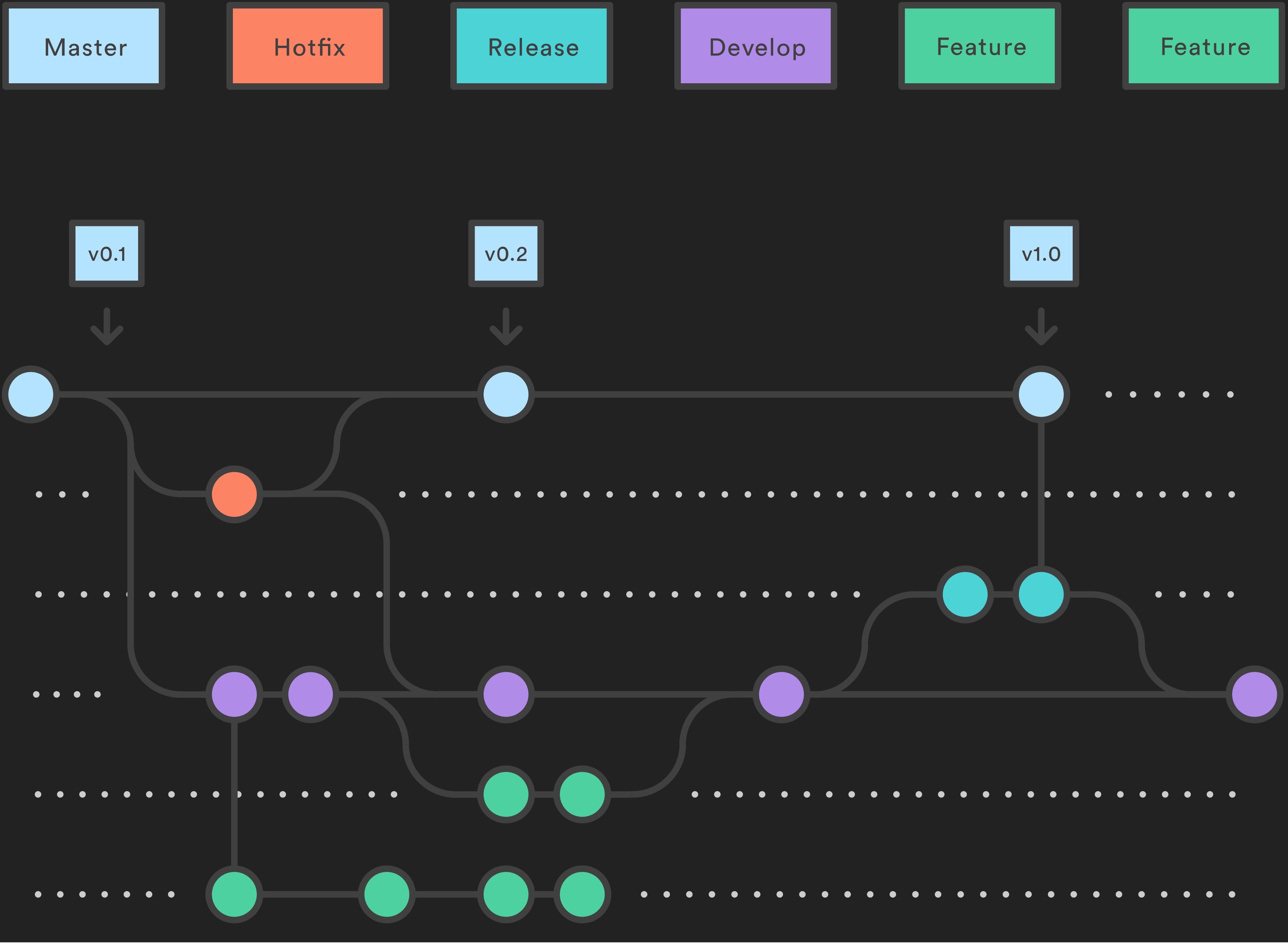

Master
Develop
Feature
Feature
v0.1
v0.2
v1.0

HOTFIX BRANCHES

Hotfixes are branched from master to be fixed and then merged back into Master and Development branches.

Speaker notes

Maintenance or “hotfix” branches are used to quickly patch production releases. Hotfix branches are a lot like release branches and feature branches 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.


Master
Hotfix
Release
Develop
Feature
Feature
v0.1
v0.2
v1.0

THE OVERALL FLOW OF GITFLOW IS:

- A develop branch is created from master
- A release branch is created from develop
- Feature branches are created from develop
- When a feature is complete it is merged into the develop branch
- When the release branch is done it is merged into develop and master
- If an issue in master is detected a hotfix branch is created from master
- Once the hotfix is complete it is merged to both develop and master

RESOURCES

- More detail on this workflow
- A comparision of common workflows