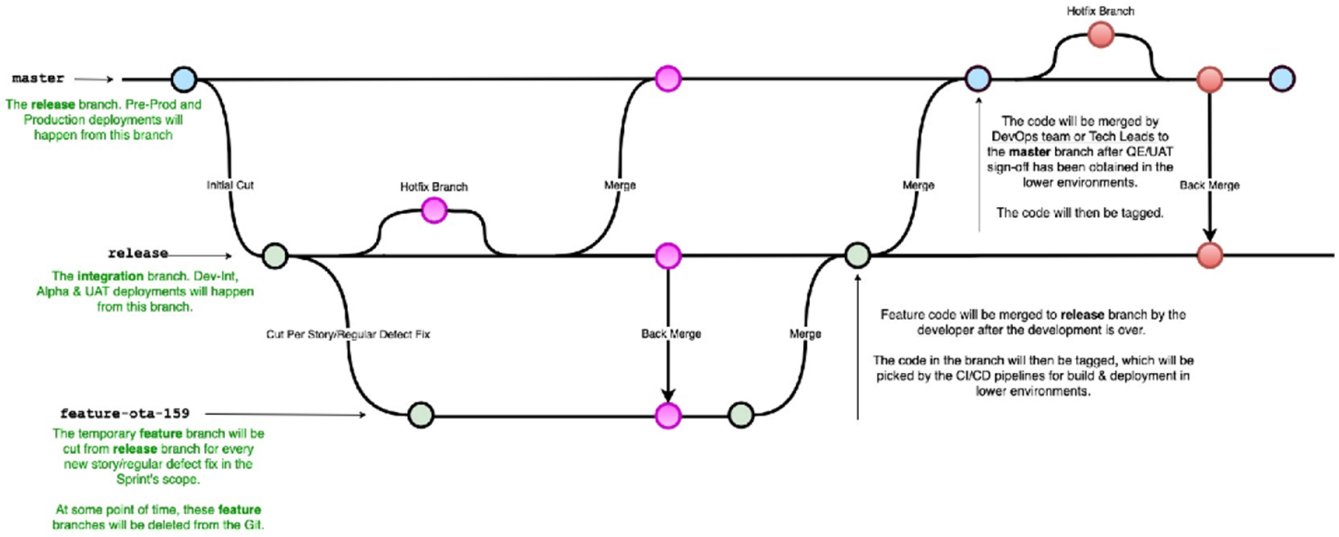
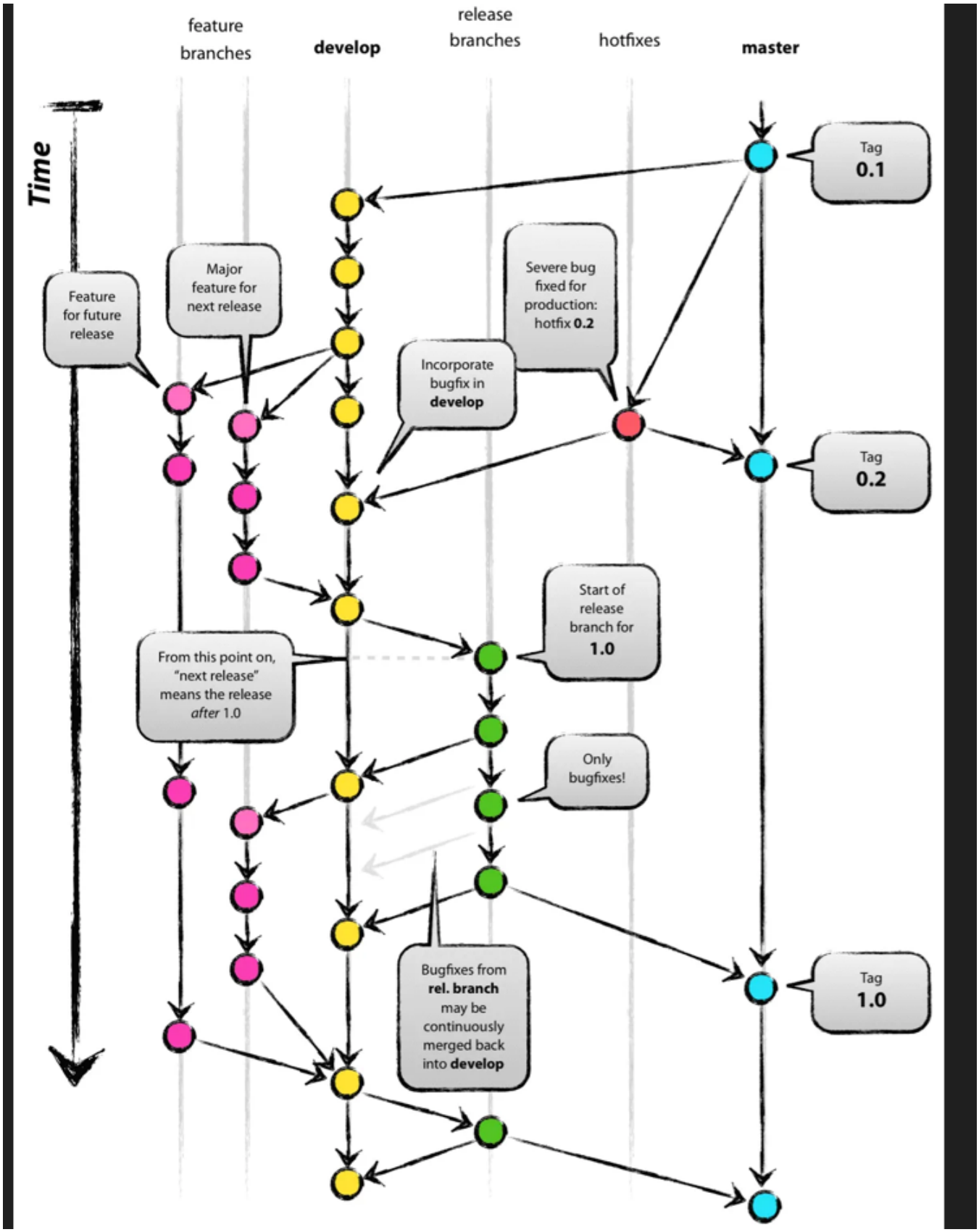
Release Branch Strategy

Single branch Strategy



Multi Branch Strategy for multiple releases


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

Master branch : `origin/master` to be the main branch where the source code of `HEAD` always reflects a *production-ready* state

Develop Branch : Its parallel to master branch. `origin/master` to be the main branch where the source code of `HEAD` always reflects a *production-ready* state.

Feature branches

blocked URL

May branch off from: develop

Must merge back into: develop

Release branches

May branch off from: develop

Must merge back into: develop and master

Branch naming convention: release-*

Release branches support preparation of a new production release. They allow for minor bug fixes and preparing meta-data for a release (version number, build dates, etc.). By doing all of this work on a release branch, the `develop` branch is cleared to receive features for the next big release.

Hotfix branches

blocked URL

May branch off from: master

Must merge back into: develop and master

Branch naming convention: hotfix-*

When a critical bug in a production version must be resolved immediately, a hotfix branch may be branched off from the corresponding tag on the master branch that marks the production version.