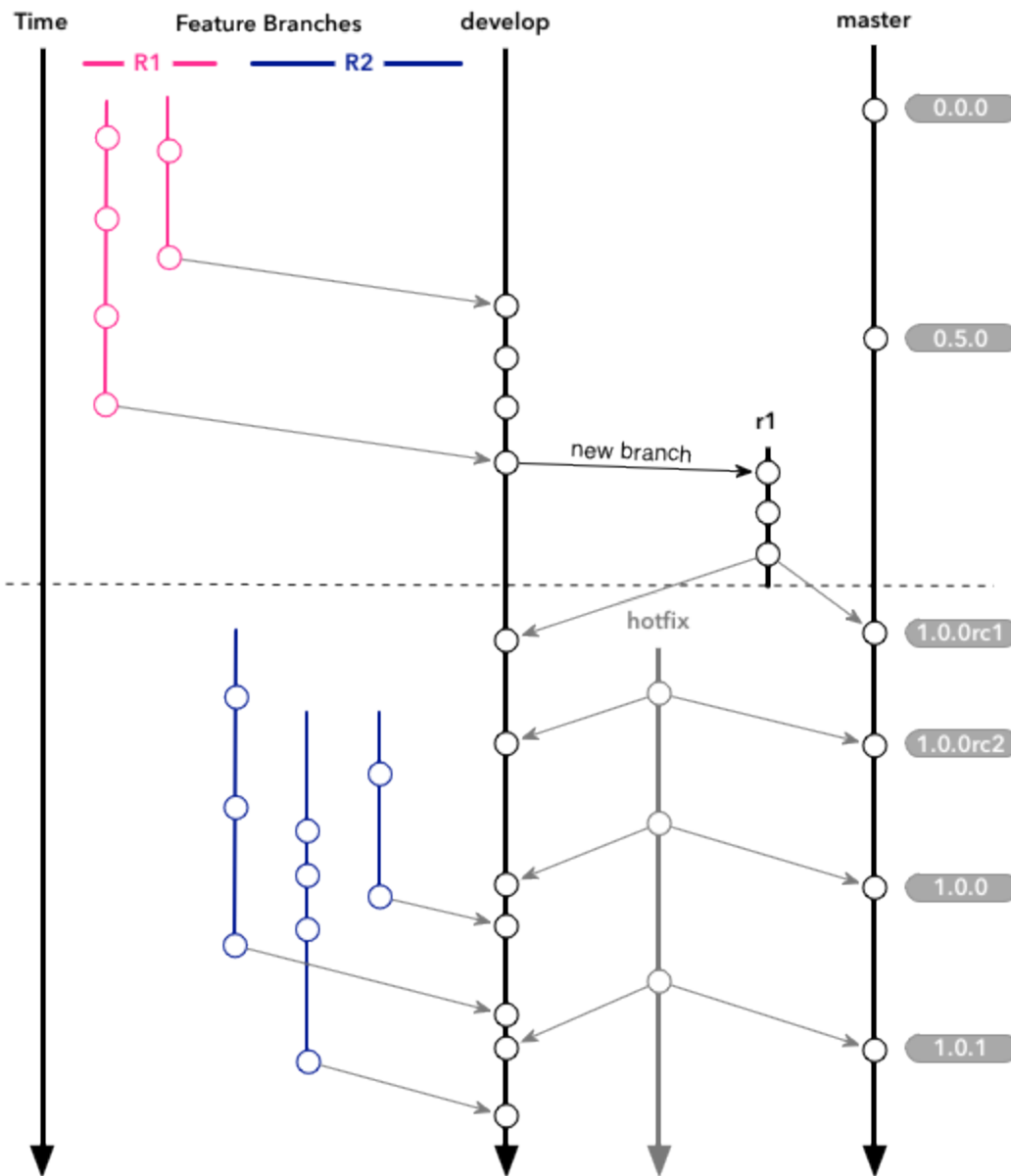


Branching and Tagging Releases

Avalon Branching & Tagging



Pre-Release Development Cycle

1. **master** is renamed **develop**
2. **release/1.0.0** is created off of **develop** and used only for **r1**-related bugfixes.
3. Features bound for post-R1 releases are merged only into **develop**.
4. When R1 is ready, **release/1.0.0** is merged into **master**. (It may actually overlap the **rcX** releases a bit.)
5. 1.x release candidates and releases are tagged on **master**.

Normal Development Cycle

1. New features (destined for 1.1.0 or beyond) are developed in branches named after the story ticket and *very* short description (e.g., **feature/VOV-XXXX-description**).
2. Completed features are merged into **develop** and their **feature** branches deleted. Any developer can merge, but a second set of eyes is encouraged.
3. Fixes that need to be released ASAP (i.e., security issues or things that severely impact existing advertised functionality) are developed in **hotfix/VOV-XXXX**.
4. Completed hotfixes are double-merged into **master** and **develop**. This *requires* the participation of the Release Manager responsible for the current **master** version.
5. As soon as a hotfix is merged:
 - a. Fixes and relevant caveats are documented in a cumulative Release Notes document, most recent changes at the top.
 - b. A release announcement/notification is sent out to avalon-l and any other appropriate lists (hydra-partners, hydra-tech, hydra-releases, etc.).
6. At code freeze for 1.1/2.0/etc., **develop** is branched as **release/x.y.z** and we go back to the Pre-Release cycle above.

[Source](#)