



Branching Policies

Marc Rosell Hernandez

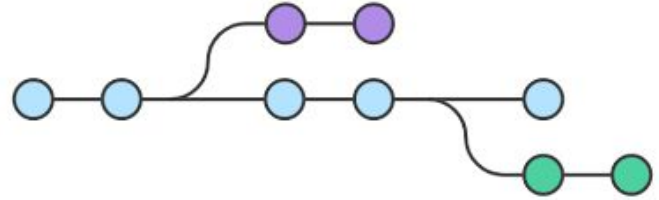


Table of Contents

- Concept
- Trunk-Based Development
- Feature Branching Development
- Gitflow Structure
- AppVeyor, Jenkins, Travis CI & others
- Homework
- Sources

Concept

- Parallel Development
- Set of Rules
 - Maintain Control
 - Code Quality and Management
- Trunk-Based Dev
- Feature Branching Dev





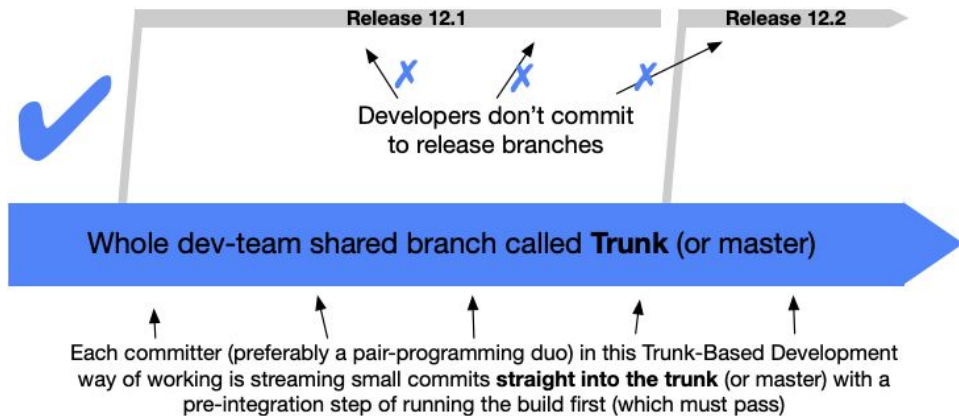
Trunk-Based Development

- Use of 2 branches
 - Dev
 - Release
- If there is an error, all team stops

Trunk-Based Development

For Smaller Teams

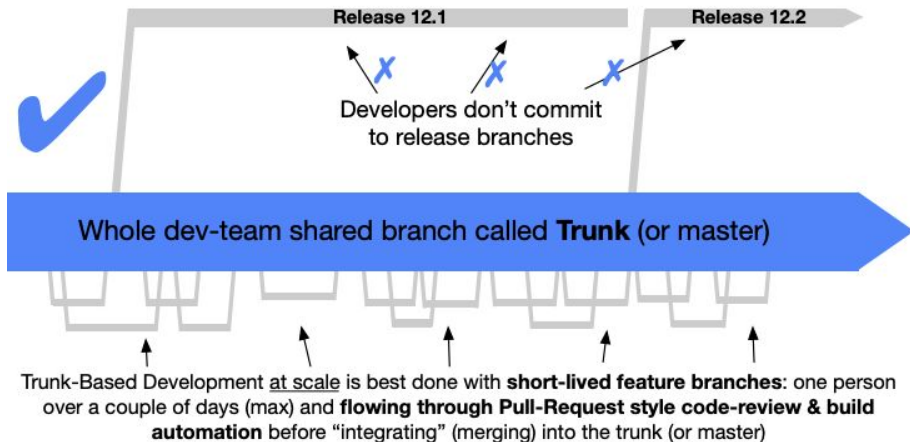
- Commit into the trunk
- Pre-integration step



Trunk-Based Development

Scaled

- Short-lived feature branches
- Pull-request style review





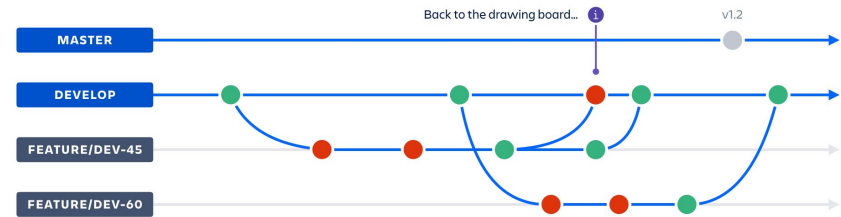
Feature Branching Development

- Feature external to the main branch
- Need to be updated
- Gitflow

Feature Branching Development

- Protect Main Branches
- Master
- Develop
- Feature

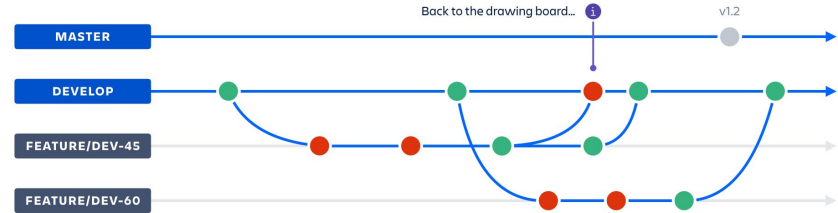
Gitflow



Feature Branching Development

- Branch Administrators
 - Lead Programmer
 - QA Lead
- Branch Off From/Merge Back
- QA Testing?

Gitflow





Feature Branching Development

Extra Rules

- Github
 - Settings>Branches>Add Rule
- Require review before merging
- ...

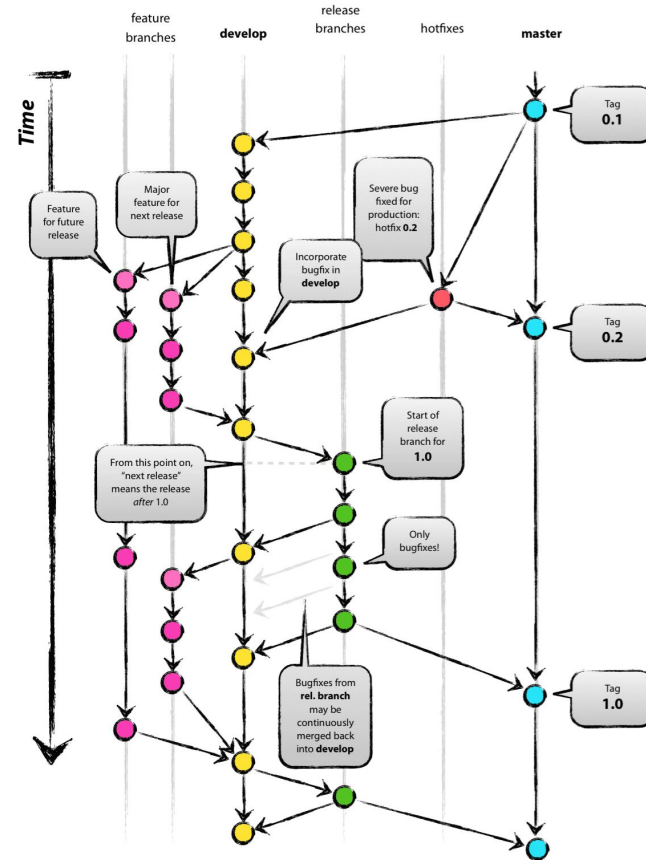
Branch protection rule

Branch name pattern
<input type="text"/>
Protect matching branches
<input type="checkbox"/> Require pull request reviews before merging When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule.
<input type="checkbox"/> Require status checks to pass before merging Choose which status checks must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.
<input type="checkbox"/> Require signed commits Commits pushed to matching branches must have verified signatures.
<input type="checkbox"/> Require linear history Prevent merge commits from being pushed to matching branches.
<input type="checkbox"/> Include administrators Enforce all configured restrictions above for administrators.
Rules applied to everyone including administrators
<input type="checkbox"/> Allow force pushes Permit force pushes for all users with push access.
<input type="checkbox"/> Allow deletions Allow users with push access to delete matching branches.
Create

Gitflow Structure

Branches:

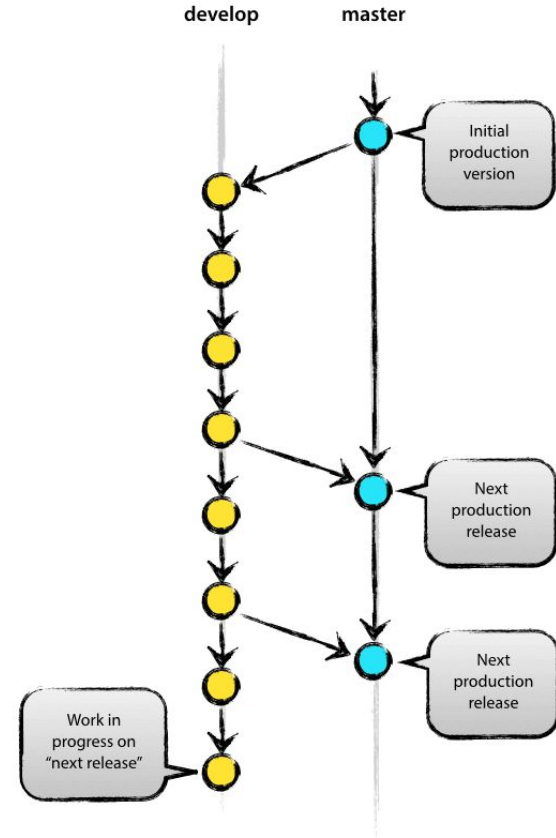
- Master
- Develop
- Feature
- Release
- Hotfix



Gitflow Structure

Master and Develop Branches

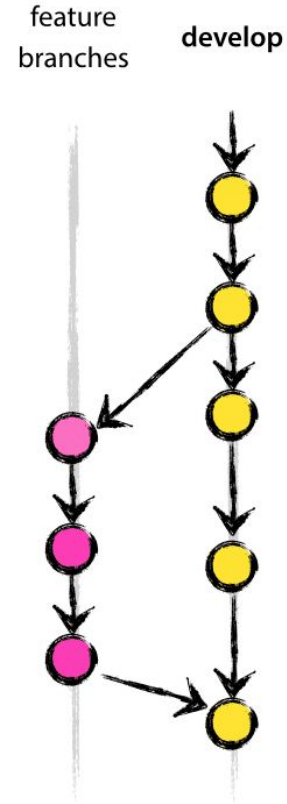
- From start to end of the project
- Master updated with main stable versions



Gitflow Structure

Feature Branches

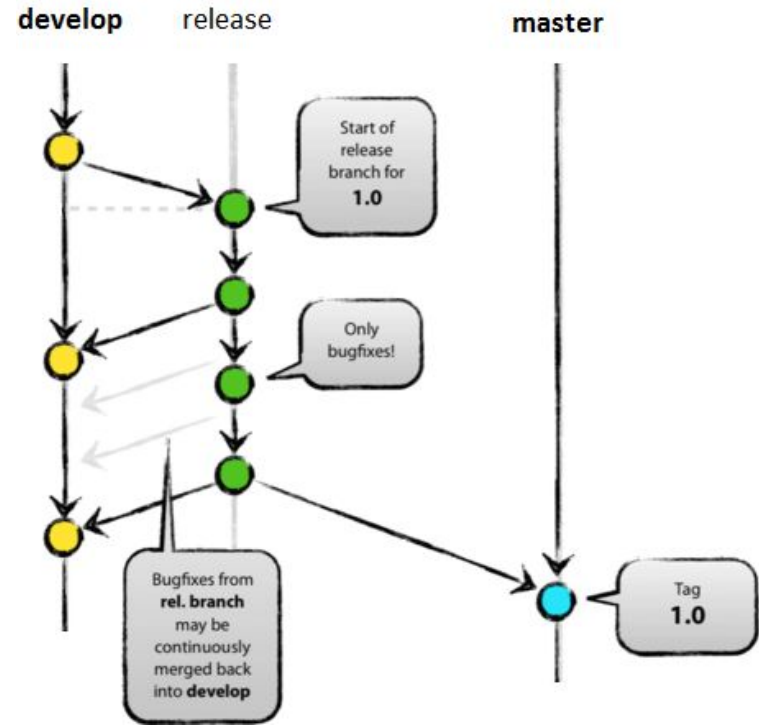
- Everyone on their own
- Keep updated
- Branch off from: **develop**
- Merge back into: **develop**



Gitflow Structure

Release Branch

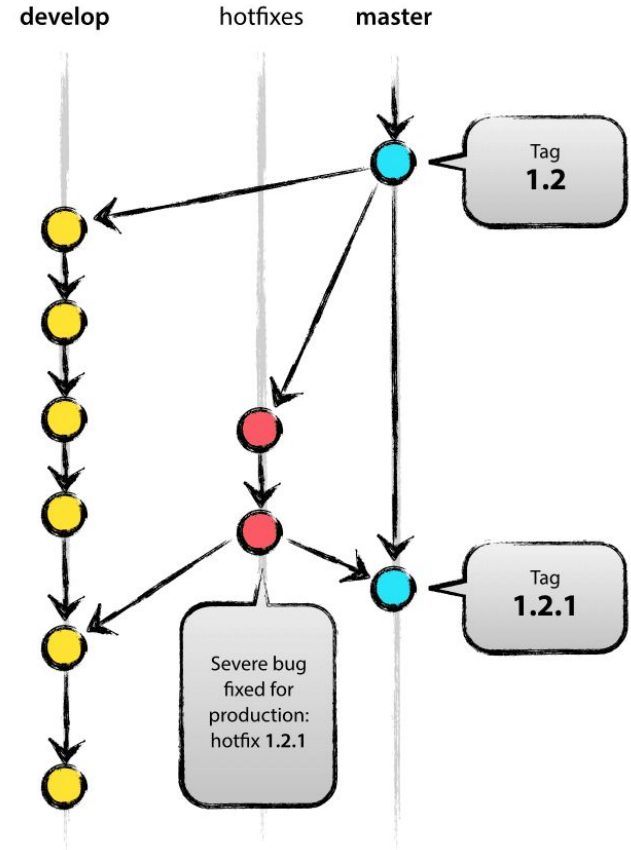
- QA Testing
 - If bug solved merge to develop
- Branch off from: **develop**
- Merge back into: **develop and master**



Gitflow Structure

Hotfix Branch

- Create when critical bug occurs in master
- Devs can keep working on features
- Branch off from: **master**
- Merge back into: **develop and master**





AppVeyor, Jenkins, Travis CI & others

- Generate Build Versions of the Project
- Commit and Test





Homework

- In groups, structure your project using the Gitflow structure, decide the administrators of each branch and take a look at Github's Branches optional rules.



Sources

- <https://nvie.com/posts/a-successful-git-branching-model/>
- <https://help.github.com/en/github/administering-a-repository/about-protected-branches>
- <https://www.appveyor.com/docs/branches/>
- <https://www.atlassian.com/git/tutorials/comparing-workflows>
- <https://docs.microsoft.com/en-us/azure/devops/repos/git/branch-policies?view=azure-devops>
- <https://docs.microsoft.com/en-us/azure/devops/repos/git/pull-requests?view=azure-devops&tabs=new-nav#complete-the-pull-request>
- <https://help.github.com/en/github/collaborating-with-issues-and-pull-requests>
- <https://devopsnet.com/2012/11/01/exciting-branching/>



Sources

- <https://git-scm.com/book/en/v2/Git-Branching-Branching-Workflows>
- <http://ithare.com/version-control-branching-for-gamedev/>
- <https://jenkins.io/doc/>
- <https://docs.travis-ci.com/>
- <https://trunkbaseddevelopment.com/>