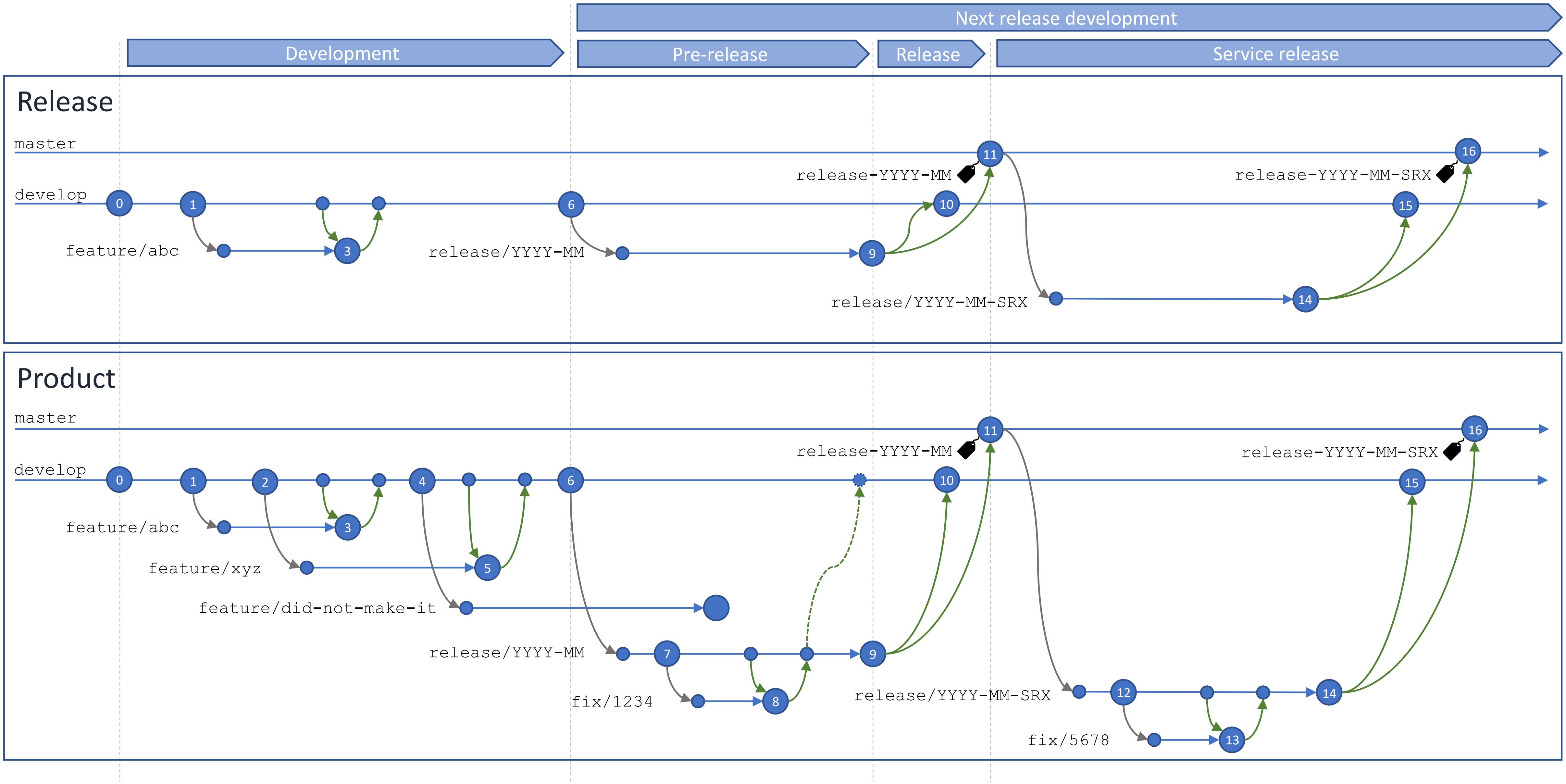
Release Engineering Flow



Release Engineering Reference

Step	Description	Git commands
Development		
0	Release development starts on the develop branch by updating the release number, upgrading to newer versions of third-party dependencies, updating the target platform definitions. Large third-party dependencies may be mirrored/downloaded if necessary.	

1, 2	Development of major features is done on feature branches of affected repositories. If such features introduce third-party dependencies, then a feature branch is created in the release repository as well. A release feature branch can also be created in order to execute an end-to-end feature build and deploy binary artifacts for testing before the feature branch is merged into develop.	`git checkout -b feature/abc develop`
		`git push -u origin feature/abc`

3, 5	Once a feature is implemented it is merged into develop. First, develop is merged into the feature branch and any merge conflicts and other problems are resolved. Then the feature branch is merged into develop with --no-ff option is order to preserve branching history. Once merged, the feature branch is deleted. The pipeline build is executed. The feature branch directory on the web server is deleted.	`git merge develop`
		`git push`
		`git checkout develop`
		`git merge --no-ff feature/abc`
		`git push`
		`git branch -d feature/abc`
		`git push origin -delete feature/abc`

4	A feature branch may be created in the same way as 1 and 2, but it might not make it into the release, i.e. merged into develop before 6. In this case the feature is moved to the next release.	
Pre-release		

6	A release branch is created off develop in release and affected product repositories. If there were no changes in some products then the pipeline definition in the release branch in the release repository shall be updated to point to the last release tag for those products. However, it is better to create release branches in all products for consistency and also to account for a possibility of a bug fix in a product which was not modified during the develop phase.	`git checkout -b release/YYYY-MM develop`
		`git push -u origin release/YYYY-MM`

7, 8	Simple fixes can be done on the release branch. Complex fixes are done on fix branches which are merged into the release branch in the same way as features – see steps 1 and 3. Fixes can be merged into the develop branch as they are closed.	
9	Before merging the release branch into develop and master the semantic versions of products are updated to reflect the nature of changes implemented as part of the release.	
10	After merging the pipeline definition is modified to point to the release parameter branch is it was modified before in step 6 to point to release tags. An end-to-end build off develop is executed.	
Release		

11	Release branch is merged into master (see 3,5), the pipeline definition is updated if it was changed in step 6 (see 10), a tag is created, an end-to-end build off master is executed, the release branch is deleted (see 3,5).	`git tag release-YYYY-MM`
		`git push origin release-YYYY-MM`

Service Release		
12 - 16	Service release is branched off master release tag and follows the same pattern as a release – only fixes are implemented.	