Feature Flow

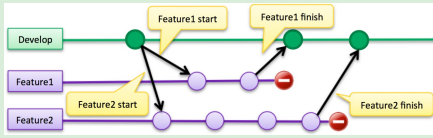## Feature Flow - Maven JGitFlow Plugin

Feature start

mvn clean jgitflow:feature-start

-DfeatureName=NewFeature

-DenableFeatureVersions=true

-DallowSnapshots=true

Feature finish

mvn clean jgitflow:feature-finish

-DfeatureName=NewFeature

-DenableFeatureVersions=true

-DallowSnapshots=true

Feature Flow - Manual Git

🔲 Feature start

git checkout develop

git pull origin develop

git checkout -b feature/*nomefeature*

manually update your code version to branch naming conventions

🔲 Feature finish

*starting from feature/*nomefeature* branch*
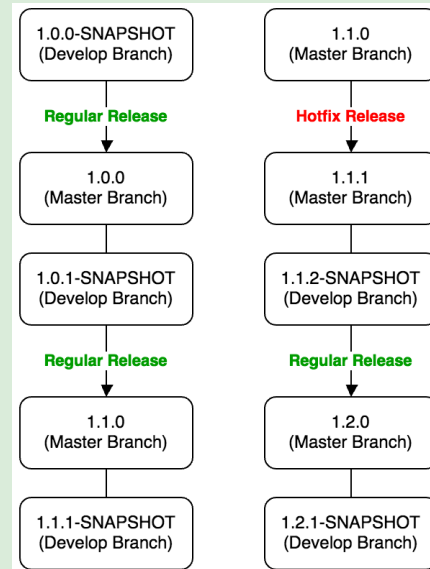
git commit -am"your feature comment here"

git checkout develop

git merge feature/*nomefeature*

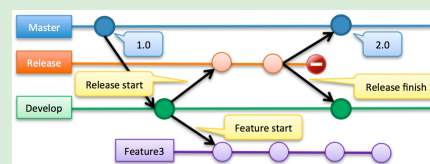*manually update your code version to develop naming conventions*

git branch -D feature/*nomefeature*

Release Version conventions

1.0.0-SNAPSHOT (Develop Branch)	1.1.0 (Master Branch)
Regular Release	**Hotfix Release**
1.0.0 (Master Branch)	1.1.1 (Master Branch)
1.0.1-SNAPSHOT (Develop Branch)	1.1.2-SNAPSHOT (Develop Branch)
Regular Release	**Regular Release**
1.1.0 (Master Branch)	1.2.0 (Master Branch)
1.1.1-SNAPSHOT (Develop Branch)	1.2.1-SNAPSHOT (Develop Branch)

Release Flow

Release Flow - Maven JGitFlow Plugin

Release start

mvn clean jgitflow:release-start

-DautoVersionSubmodules=true

-DallowSnapshots=true

-DallowUntracked=false

-DpushReleases=false

-DupdateDependencies=true

Release finish

mvn clean jgitflow:release-finish

-DpushReleases=true -DnoDeploy=true

-DautoVersionSubmodules=true

-DallowSnapshots=false

Release Flow - Manual Git

🔲 Release start

git checkout develop

git pull origin develop

git checkout -b release/*release-version*

manually update your code version to release branch naming conventions

🔲 Release finish

*starting from release/*release-version* branch*

git commit -am"your release changes comment here"

git pull origin release/*release-version*

git checkout develop

git pull origin develop

git merge release/*release-version*

manually update your code version to stable version naming conventions

git commit -am"*release-version* "

git checkout master

git pull origin master

git merge release/*release-version* --squash

git commit -am"*release-version* "

git tag *release-version*

git push origin master --tag

git checkout develop

manually update your code version to next develop version naming conventions

git commit -am"*next-version*"

git push origin develop

git branch -D release/*release-version*

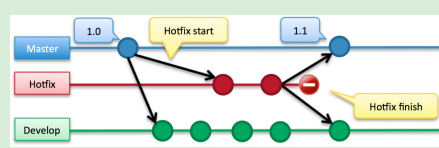### Hofix Simple Flow

Hotfix Simple Flow - Maven JGitFlow Plugin

Hotfix start

mvn clean jgitflow:hotfix-start

-DautoVersionSubmodules=true

-DallowSnapshots=true

-DpushHotfixes=false

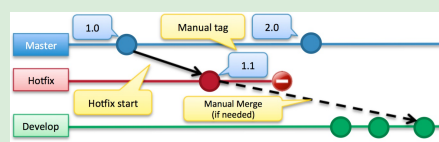**-DallowUntracked=false**

-DupdateDependencies=true

Hotfix finish

mvn clean jgitflow:hotfix-finish

-DpushHotfixes=true -DnoDeploy=true

-DautoVersionSubmodules=true

-DallowSnapshots=false

Hotfix Simple Flow - Manual Git

Same as Release one but with hotfix naming conventions

Hofix Complex Flow

Since a newer stable is present on master branch, tag and branch cannot be merged back.

Hotfix Complex Flow - Maven JGitFlow Plugin

Hotfix start: same as simple.

Hotfix finish: cannot be managed, use manual flow

Hotfix Complex Flow- Finish - Manual Git

starting from hotfix/hotfix-version branch

git commit -am"your hotfix changes comment here"

git pull origin hotfix/*hotfix-version*

manually update your code version to stable version naming conventions
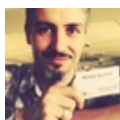
git commit -am"*hotfix-version*"

git tag *hotfix-version*

git push origin --tag

git checkout develop

manually merge hotifx changes to develop if applicable

git branch -D hotfix/*hotfix-version*