**Source Tree Merging Branches**

**Branches**

**Production**

**Release**

**Development**

**Features**

**Hot Fix**

**Summary of the concept**

The general idea of git-flow is to use the following branch structure in your repository:

* **Development branch** (usually called ‘develop’)  
  This is your main development branch where all the changes destined for the next release are placed, either by directly committing small changes or by merging other branches (e.g. feature branches) into this branch.
* **Production branch** (usually called ‘master’)  
  This branch represents the latest released / deployed codebase. Only updated by merging other branches into it.
* **Feature branches** (usually prefixed with ‘feature/’)  
  When you start work on anything non-trivial, you create a feature branch. When finished, you’ll merge this branch back into the development branch to queue it for the next release.
* **Release branches** (usually prefixed with ‘release/’)  
  When you’re about to package a new release, you create a release branch from the development branch. You can commit to it during your preparation for a release, and when it’s ready to be deployed, you merge it into both the development branch and the master branch (to indicate that the release has been deployed).
* **Hotfix branches** (usually prefixed with ‘hotfix/’)  
  If you need to patch the latest release without picking up new features from the development branch, you can create a hotfix branch from the latest deployed code in master. Once you’ve made your changes, the hotfix branch is then merged back into both the master branch (to update the released version) and the development branch (to make sure the fixes go into the next release too)

@https://blog.sourcetreeapp.com/2012/08/01/smart-branching-with-sourcetree-and-git-flow/