

Chapter 1 - Introduction to GitFlow

Objectives

Key objectives of this chapter

- Understanding GitFlow
- Understanding Features
- Understanding Releases
- Understanding Hotfixes

1.1 What is GitFlow

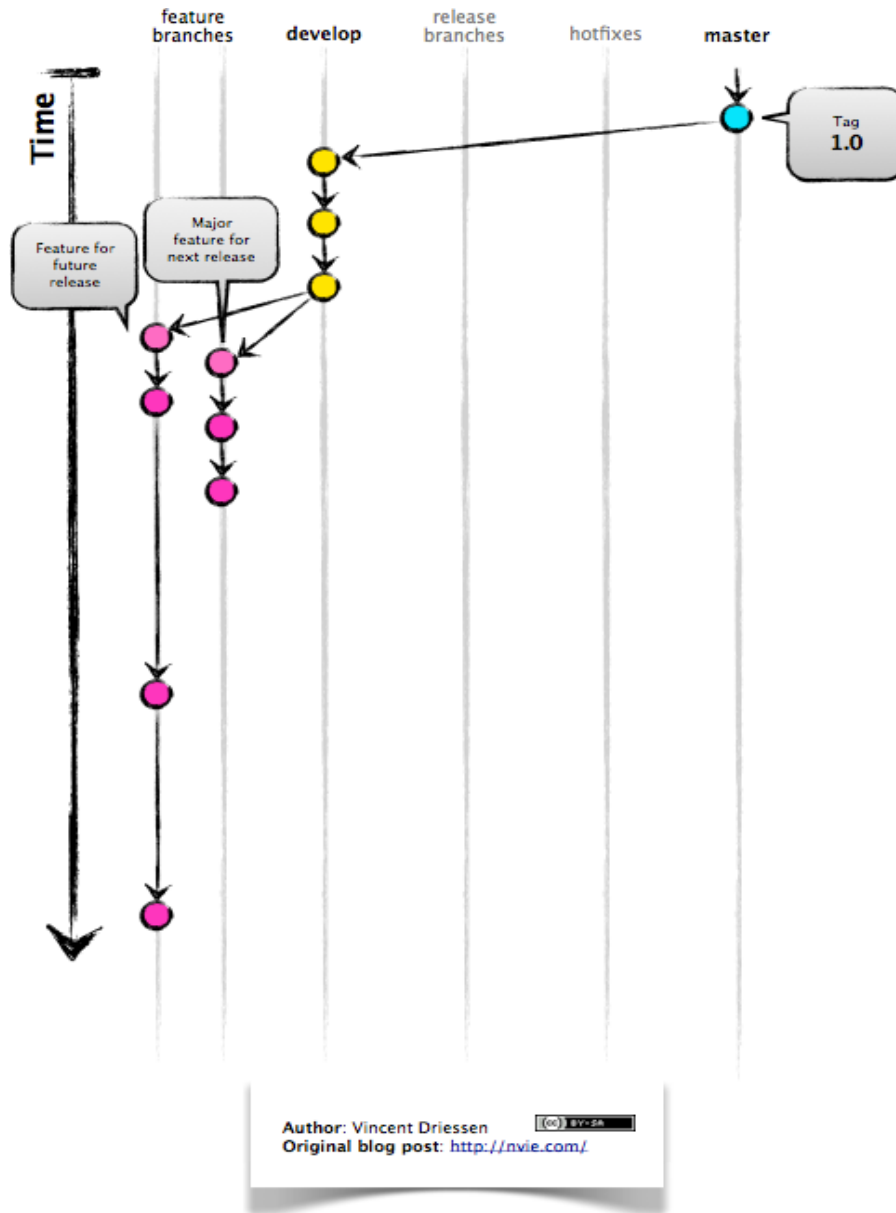
- GitFlow is a branching model for Git.
- Created by Vincent Driessen
- A set of git extensions to provide high-level repository operations.
- It's model is scalable.
- Well suited to collaboration

1.2 Benefits

- Parallel development
 - ◇ Isolates new development work from finished work
 - ◇ Code is merged back into main body of code when developers are happy that code is ready for release.
- Collaboration
- Release Staging Area
 - ◇ New development work is merged back into the develop branch
 - ◇ 'Develop' branch is a staging area for all completed features that haven't yet been released.
- Support for hotfixes
 - ◇ Hotfix branches – branches made from a tagged release which can be used to make an emergency change.

1.3 How GitFlow works?

- New development are built in **feature** branches.



Canada

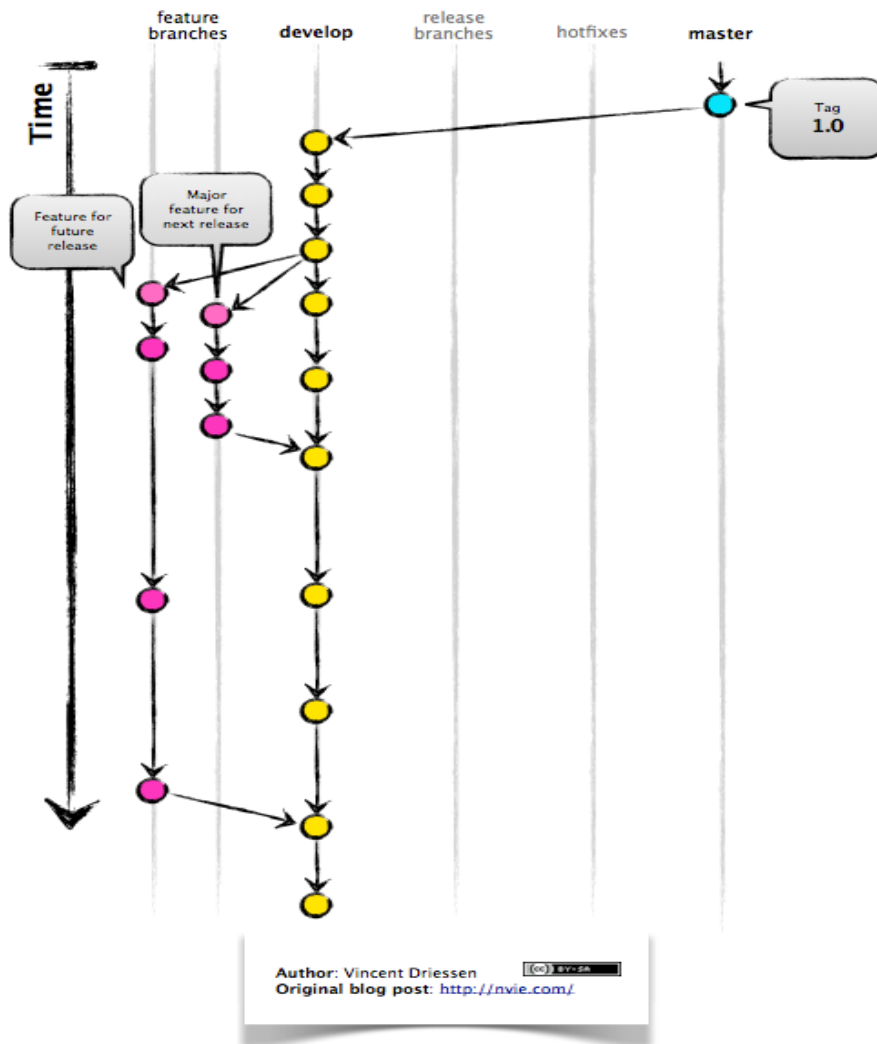
821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

436 YorkRoad, Suite 1
Jenkintown, PA, 19046
1 877 517 6540
getinfousa@webagesolutions.com

1.4 How GitFlow works? (Contd.)

- Feature branches are branched off of the **develop** branch.
- Finished features are merged back into the **develop** branch.



Canada

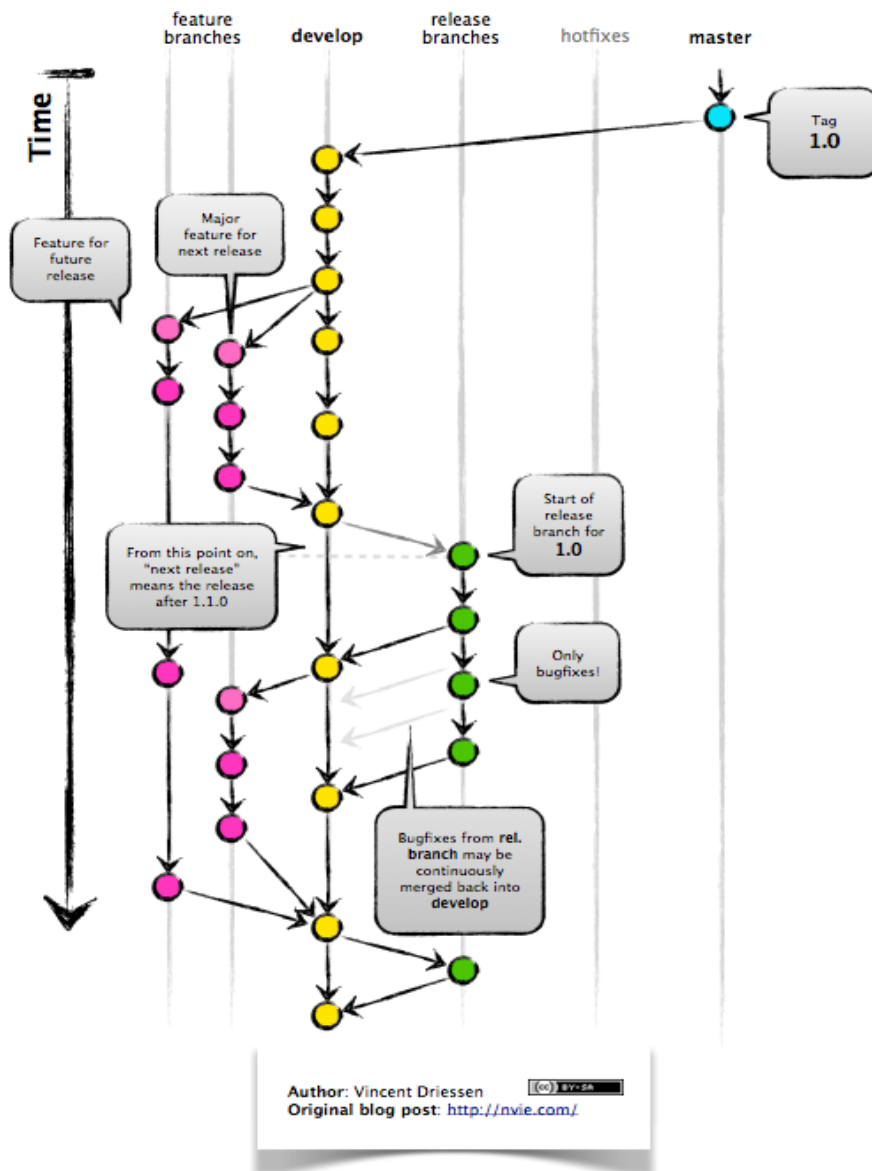
821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

436 YorkRoad, Suite 1
Jenkintown, PA, 19046
1 877 517 6540
getinfousa@webagesolutions.com

1.5 What is GitFlow? (Contd.)

- When features are completed, a **release** branch is created off of **develop** branch.



Canada

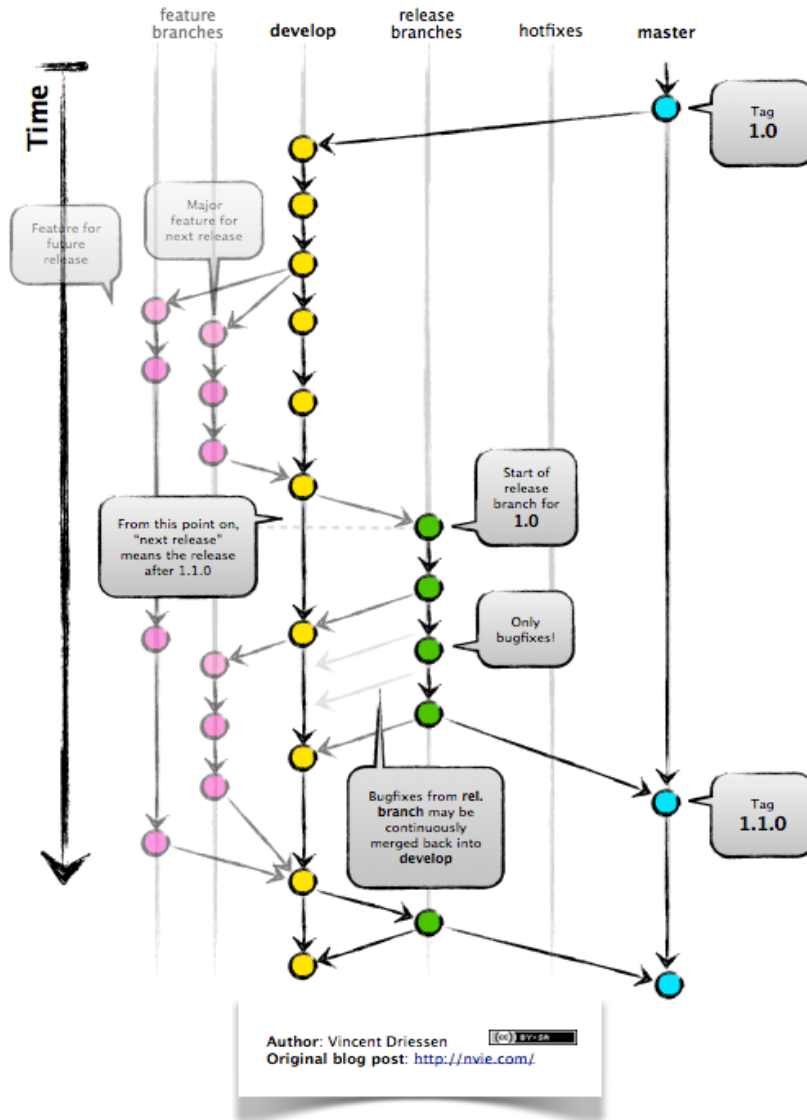
821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

436 YorkRoad, Suite 1
Jenkintown, PA, 19046
1 877 517 6540
getinfousa@webagesolutions.com

1.6 How GitFlow works? (Contd.)

- The code in the release branch is deployed, tested, fixed, redeployed, and retested.
- Release branch is merged into master and into develop branch.



Canada

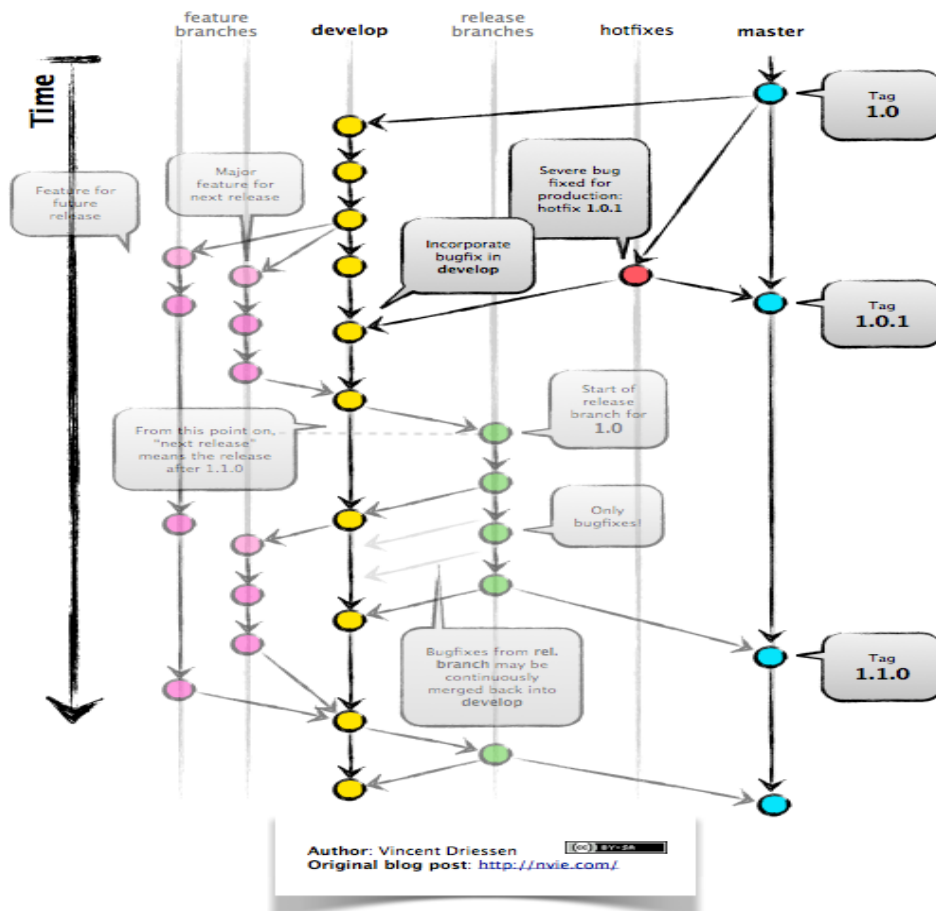
821A Bloor Street West
 Toronto, Ontario, M6G 1M1
 1 866 206 4644
getinfo@webagesolutions.com

United States

436 York Road, Suite 1
 Jenkintown, PA, 19046
 1 877 517 6540
getinfousa@webagesolutions.com

1.7 How GitFlow works? (Contd.)

- The master branch contains released code only.
- The only commits to master are merges from **release** branches and **hotfix** branches.
- Hotfix branches are branched directly from a tagged release in the master branch.
- Hotfixes, when finished, are merged back into both master and develop to ensure the hotfixes aren't accidentally lost when next release occurs.



Canada

821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

436 York Road, Suite 1
Jenkintown, PA, 19046
1 877 517 6540
getinfousa@webagesolutions.com

1.8 GitFlow Extension

- All the above can be implemented using Git's inbuilt branching and merging
- To simplify a little, there's an “extension” to Git called “GitFlow”
- The extension is shipped by default with “Git for Windows”
- Available through package managers
 - ◇ e.g. 'sudo apt-get install git-flow'

1.9 Initializing GitFlow

- Initialized inside an existing git repository

```
git flow init
```

- Recommended to use the default values for branches

1.10 Features

- New development is organized in the form features
- Typically exist in developers repository only
- Start a new feature

```
git flow feature start FEATURE_NAME
```

- Finish up a feature
 - ◇ Merges feature into develop branch
 - ◇ Removes the **feature** branch (unless you use -k switch to keep it)
 - ◇ Switches back to **develop** branch

```
git flow feature finish FEATURE_NAME
```

Canada

821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

436 YorkRoad, Suite 1
Jenkintown, PA, 19046
1 877 517 6540
getinfousa@webagesolutions.com

- Published feature
 - ◇ Useful for pushing a feature to the remote repository so it can be used by other users

```
git flow feature publish FEATURE_NAME
git flow feature pull origin FEATURE_NAME
```

1.11 Release

- Finalized features are merged into a release branch
- Allow for minor bug fixes
- Start a release
 - ◇ **Release** branch is created from the **develop** branch

```
git flow release start RELEASE [BASE]
e.g.
git flow release start REL_1.0 develop
```

- Finish up a release
 - ◇ Merges the release branch back into 'master'
 - ◇ Tags the release with its name
 - ◇ Back-merges the release into 'develop' branched
 - ◇ Removes the release branch unless you use -k switch to keep it

```
git flow release finish RELEASE -m "tag"
```

- Publish a release
 - ◇ For collaboration, a release can be published to a remote repository

```
git flow release publish RELEASE
```

Canada

821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

436 York Road, Suite 1
Jenkintown, PA, 19046
1 877 517 6540
getinfousa@webagesolutions.com

1.12 Hotfixes

- Changes required to fix an undesired state of a live production version
- Hotfixes can be branched off from the corresponding tag on the master branch that marks the production version
- Create a hotfix branch

```
git flow hotfix start VERSION [BASENAME]
```

- e.g.

```
git flow hotfix start 'REL_1.0.1' develop
```

- Finish a hotfix
- Finishing a hotfix gets merged back into **develop** and **master** branches.
- Master merge is tagged with the hotfix version

```
git flow hotfix finish VERSION
```

- e.g.

```
git flow hotfix finish REL_1.0.1
```

1.13 Summary

- GitFlow is a branching model for git
- It is a set of extensions for git.
- The branching model supports features, releases, and hotfixes.

Canada

821A Bloor Street West
Toronto, Ontario, M6G 1M1
1 866 206 4644
getinfo@webagesolutions.com

United States

436 YorkRoad, Suite 1
Jenkintown, PA, 19046
1 877 517 6540
getinfousa@webagesolutions.com