

### *A. Git branching, good practices.*

A good git branching model has primarily two branches – master and development. Master branch is generally the most stable and is untouched for regular development activities. It normally has the code that is deployed and could be tagged for different releases.

Development branch runs parallel to master and could be used for all code changes and testing. New features could be developed on a different branch that branches out from the development branch and merges back to it at a later stage (whenever feature development is done). The development branch is finally merged back into the main branch after testing. After merge, the main branch is tested again and tagged for release.

There could be other branches, for instance, a release branch which branches off from the development when the development and testing for the release are close to done. This allows the development branch to be used for the next release immediately. The release branch is merged back into both the development branch and the main branch.

A good way of branching for different users, is to use the feature branching mentioned above. Every user working on a different feature (however small) could branch out from the development branch, make the changes, test and merge back to the development branch.

### *B. Git commands related to creating, updating and deleting branches.*

Create 'dev' branch

```
git checkout -b dev
git push origin dev    //Push to github
```

On other machines, to setup the 'dev' branch

```
git pull
git checkout -t -b dev origin/dev

git add, commit..
git push origin dev    //Push to 'dev'
```

### *C. Logs while creating a new branch and pushing to github*

```
$ git checkout -b dev_0.1
```

```
$ git branch
```

```
* dev_0.1
```

```
master
```

```
$ git add hostapd-2.6/src/eap_server/eap_server_noob.c
```

```
$ git add hostapd-2.6/src/eap_server/eap_server_noob.h
```

```
$ git status
```

```
On branch dev_0.1
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
modified: hostapd-2.6/src/eap_server/eap_server_noob.c
```

```
modified: hostapd-2.6/src/eap_server/eap_server_noob.h
```

```
$ git commit -m "Hostapd changes"
```

```
[dev_0.1 266e2bc] Hostapd changes
2 files changed, 4728 insertions(+), 4380 deletions(-)
rewrite hostapd-2.6/src/eap_server/eap_server_noob.c (72%)
rewrite hostapd-2.6/src/eap_server/eap_server_noob.h (86%)
```

### **\$ git push origin dev\_0.1**

```
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 20.60 KiB | 0 bytes/s, done.
Total 7 (delta 5), reused 0 (delta 0)
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/tuomaura/eap-noob.git
80c5a5b..266e2bc dev_0.1 -> dev_0.1
$
```

*Setting up the branch on a different machine/folder.*

### **\$ git clone https://github.com/tuomaura/eap-noob.git**

```
Cloning into 'eap-noob'...
remote: Counting objects: 3930, done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 3930 (delta 8), reused 15 (delta 5), pack-reused 3899
Receiving objects: 100% (3930/3930), 7.69 MiB | 1.97 MiB/s, done.
Resolving deltas: 100% (1643/1643), done.
Checking connectivity... done.
```

### **\$ cd eap-noob/**

#### **\$ git branch**

```
* master
```

### **\$ git checkout -t -b dev\_0.1 origin/dev\_0.1**

```
Branch dev_0.1 set up to track remote branch dev_0.1 from origin.
Switched to a new branch 'dev_0.1'
```

#### **\$ git branch**

```
* dev_0.1
  master
```

### **\$ git status**

```
On branch dev_0.1
```

```
Your branch is up-to-date with 'origin/dev_0.1'.
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
    modified:   .gitignore
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

### **\$ git add .gitignore**

```
$ git commit -m "Testing - changes to gitignore"
```

```
[dev_0.1 3d40907] Testing - changes to gitignore
```

```
1 file changed, 3 insertions(+), 2 deletions(-)
```

```
$ git push origin dev_0.1
```

```
Counting objects: 3, done.
```

```
Delta compression using up to 4 threads.
```

```
Compressing objects: 100% (3/3), done.
```

```
Writing objects: 100% (3/3), 354 bytes | 0 bytes/s, done.
```

```
Total 3 (delta 2), reused 0 (delta 0)
```

```
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
```

```
To https://github.com/tuomaura/eap-noob.git
```

```
266e2bc..3d40907 dev_0.1 -> dev_0.1
```

```
$
```

*Pull changes on branch dev\_0.1,*

```
$ git pull origin dev_0.1
```

```
remote: Counting objects: 3, done.
```

```
remote: Total 3 (delta 2), reused 3 (delta 2), pack-reused 0
```

```
Unpacking objects: 100% (3/3), done.
```

```
From https://github.com/tuomaura/eap-noob
```

```
* branch      dev_0.1    -> FETCH_HEAD
```

```
266e2bc..3d40907 dev_0.1  -> origin/dev_0.1
```

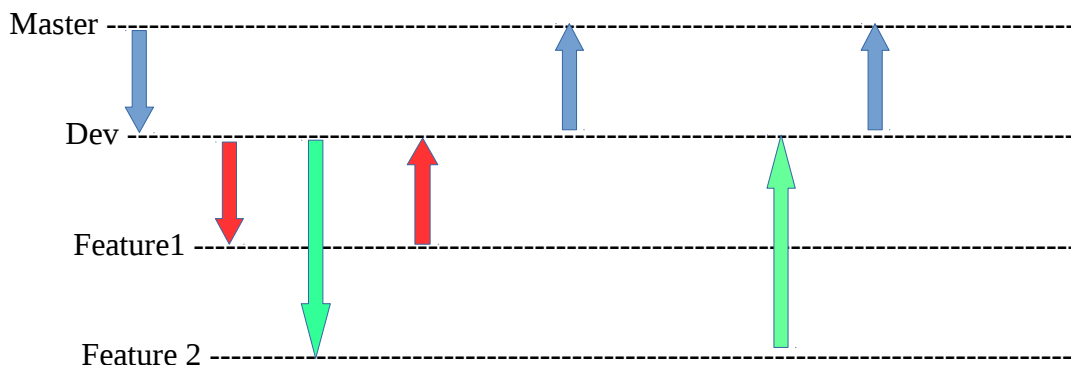
```
Updating 266e2bc..3d40907
```

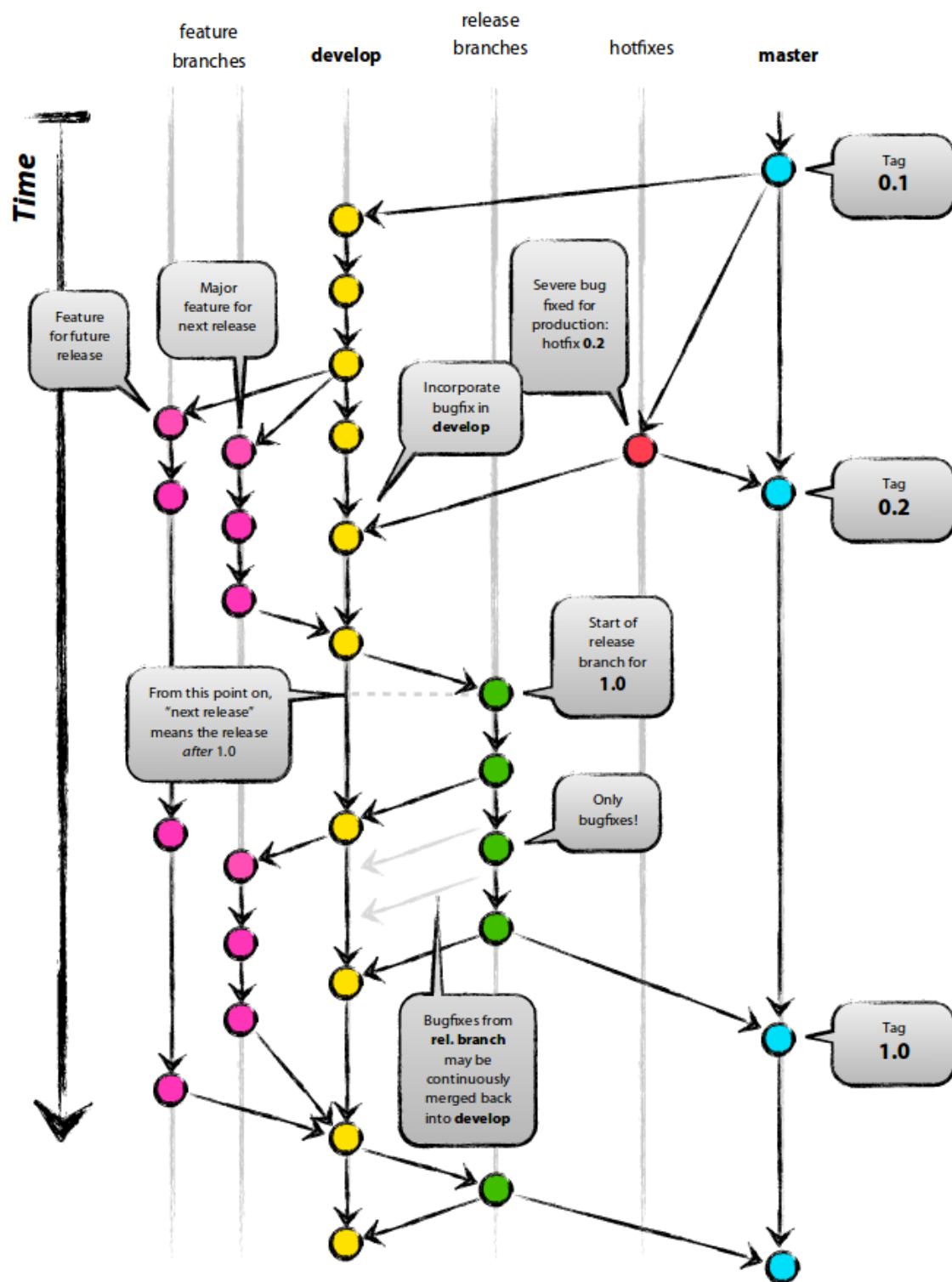
```
Fast-forward
```

```
.gitignore | 5 +++--
```

```
1 file changed, 3 insertions(+), 2 deletions(-)
```

D.





Author: Vincent Driessen  
 Original blog post: <http://nvie.com/posts/a-successful-git-branching-model>  
 License: Creative Commons BY-SA

*E. Source:*

<http://nvie.com/posts/a-successful-git-branching-model/>

[http://genomewiki.ucsc.edu/index.php/Working with branches in Git](http://genomewiki.ucsc.edu/index.php/Working_with_branches_in_Git)

<https://github.com/Kunena/Kunena-Forum/wiki/Create-a-new-branch-with-git-and-manage-branches>