

Distributed Software Control Management

Git

History

April 3, 2005 – development starts

April 6 – project announced

April 18 – first multi-branch merge

April 29 – speed = 6.7 patches / s

June 16 – tracking 2.6.12 release

Why Use Git?

File Contents, not files

Cheap branch / merge

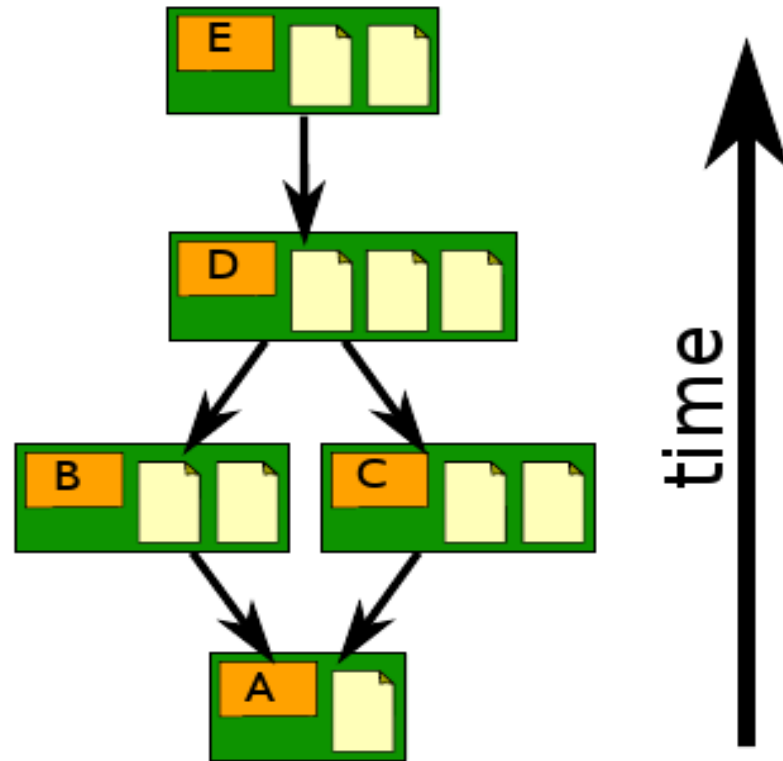
Fits Open Source

Why Not to Use Git?

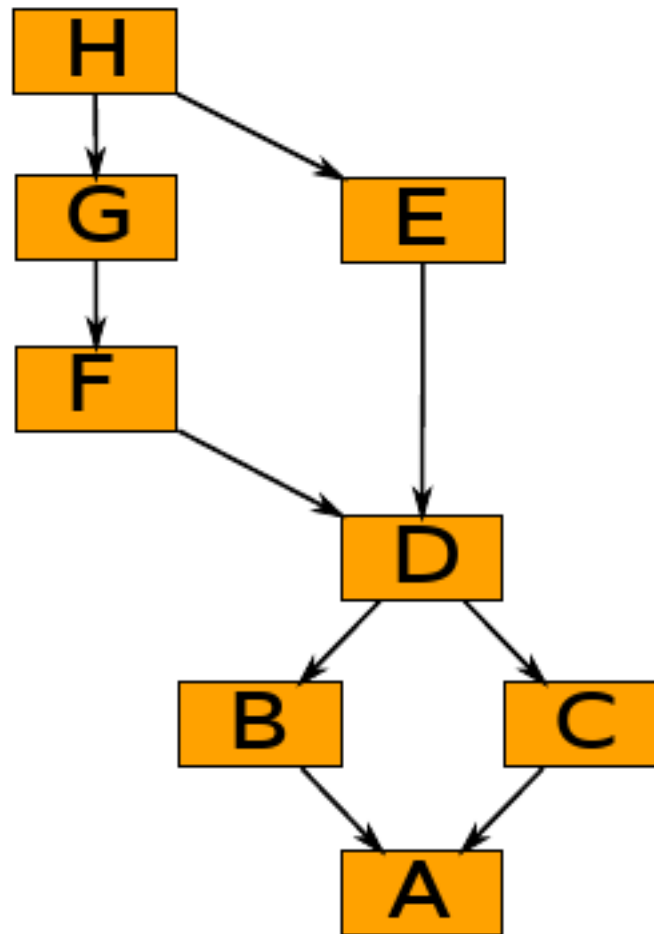
It's New

Has Steep Learning Curve

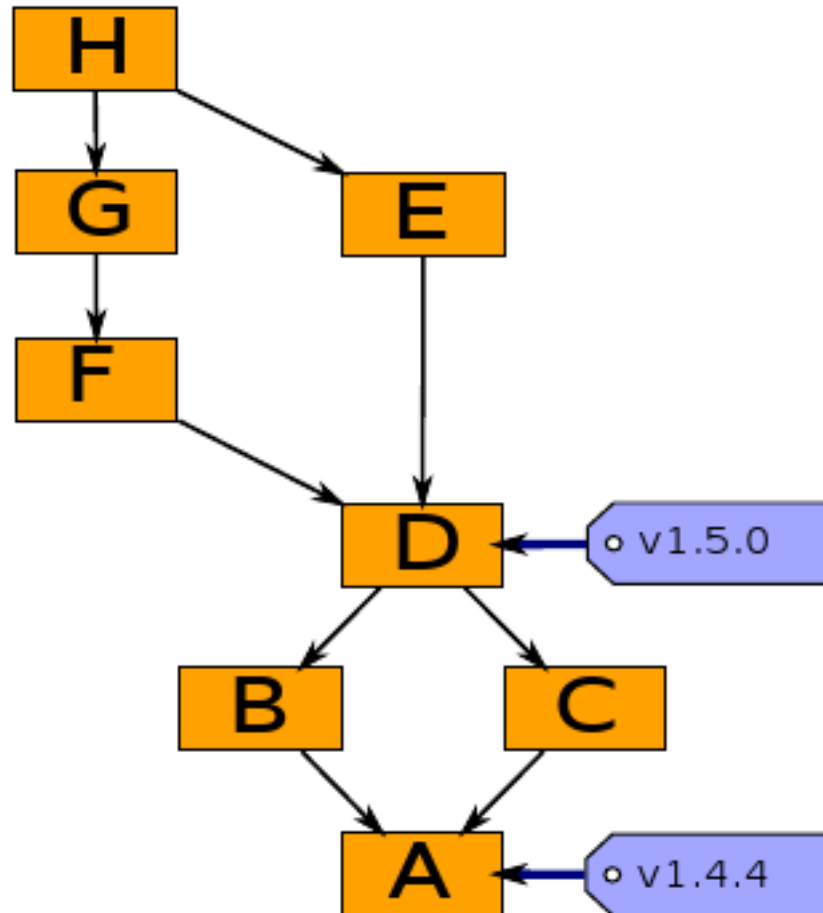
Basic Repository Operations



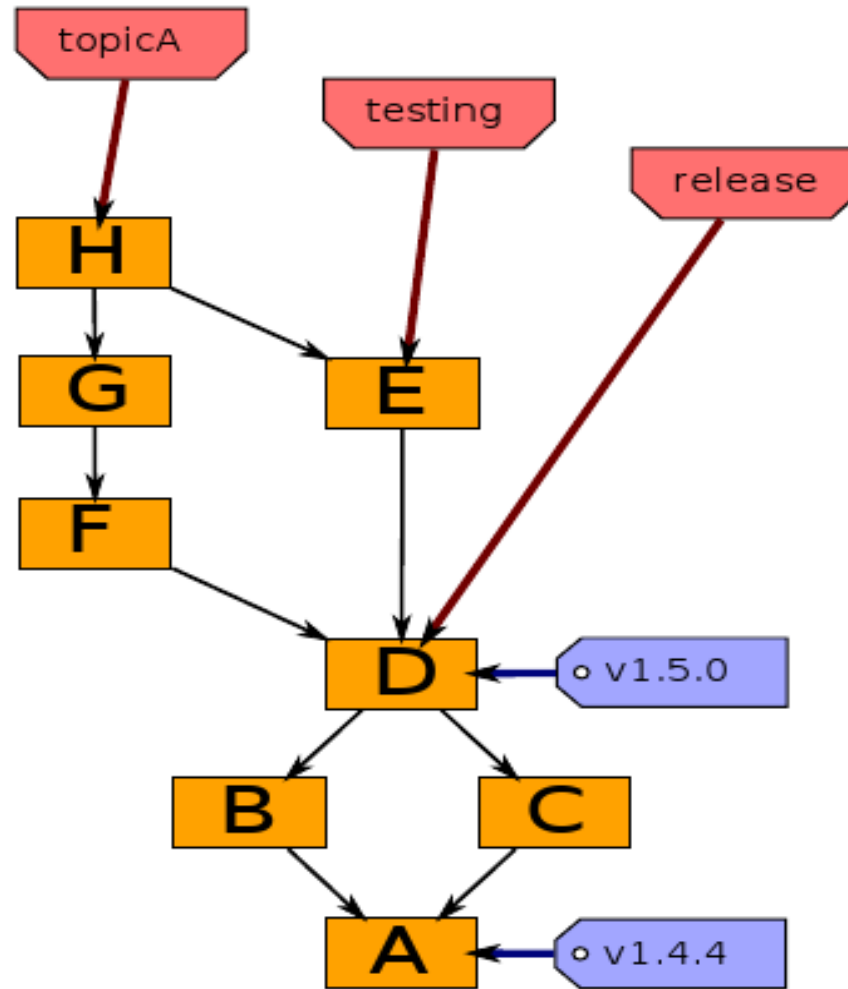
Directed Acyclic Graph

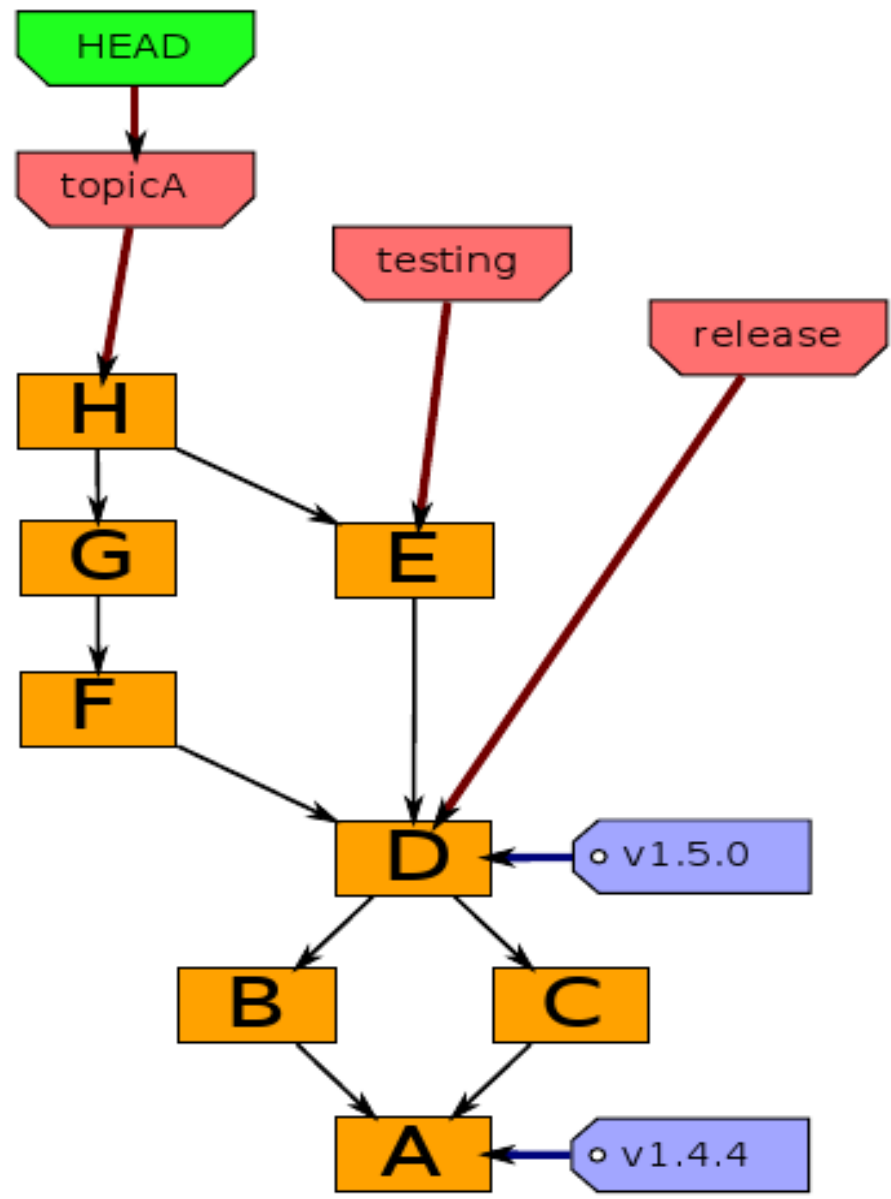


Tags



Branches





Typical Open Source Workflow

Grab someone's code

```
git clone URL
```

Get any updates from their codebase

```
git pull
```

Push any updates to their codebase

```
git push
```

Working on a Repo

Add a new file

```
git add new_file_name
```

Commit Changes

```
git commit -a -m"my log msg"
```

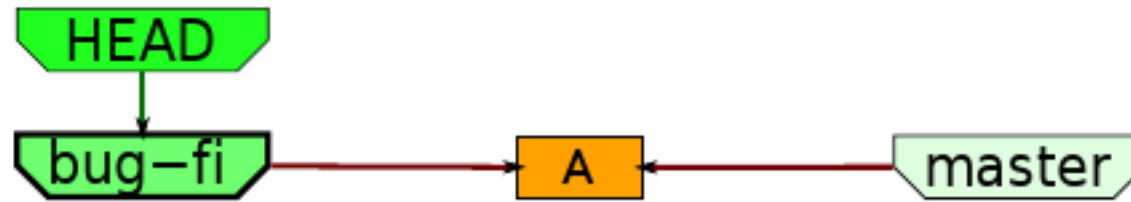
Show Changes like SVN

```
git show --name-status
```

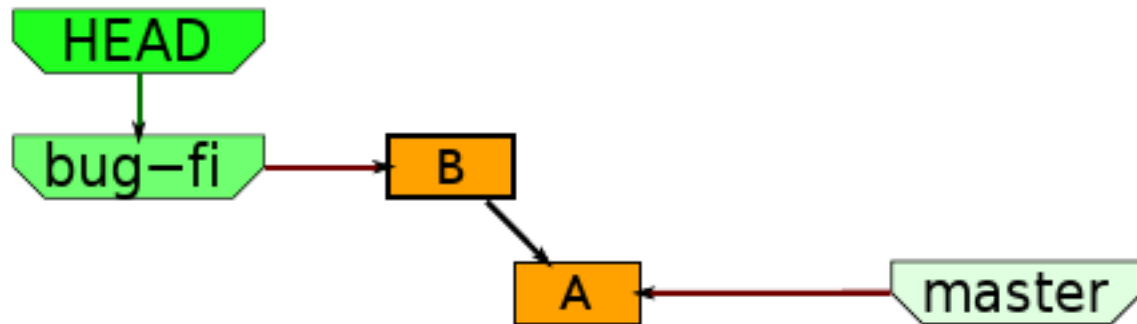
git init



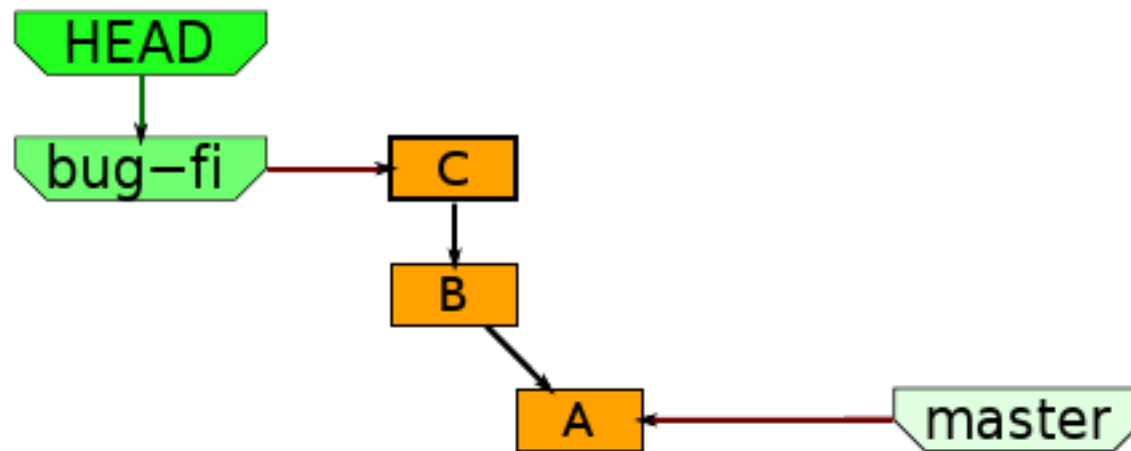
```
git checkout -b bug_fix
```



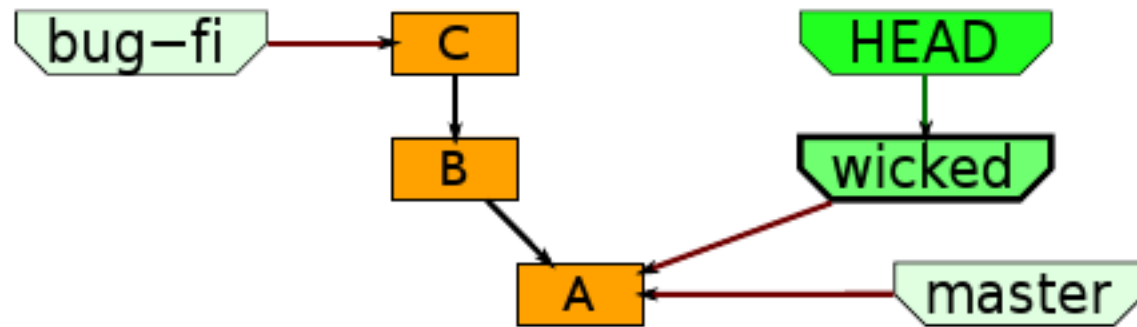
```
git commit -a -m "B"
```



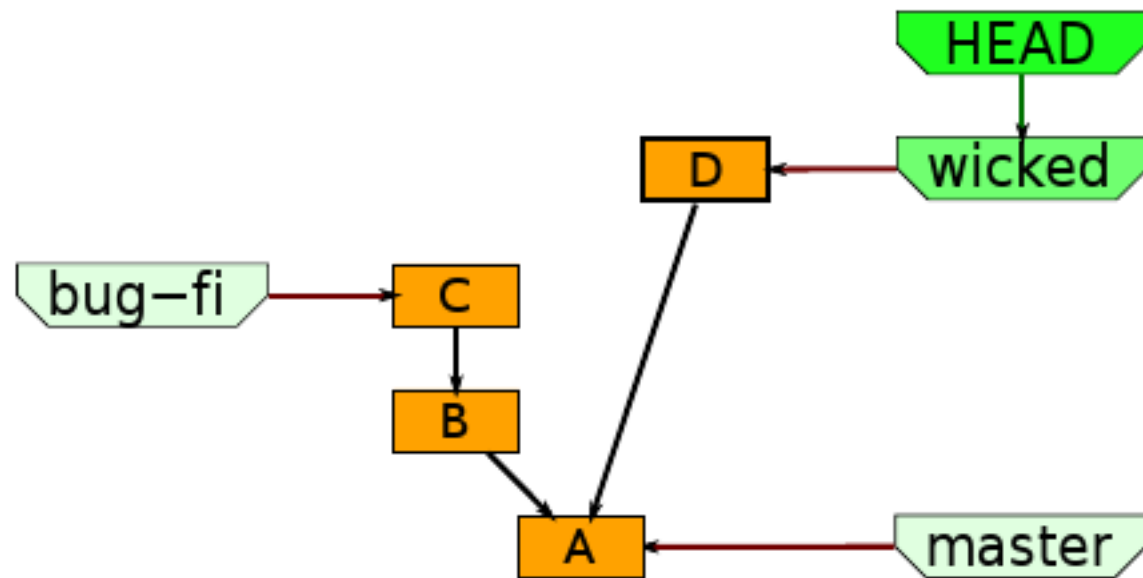
```
git commit -a -m "C"
```



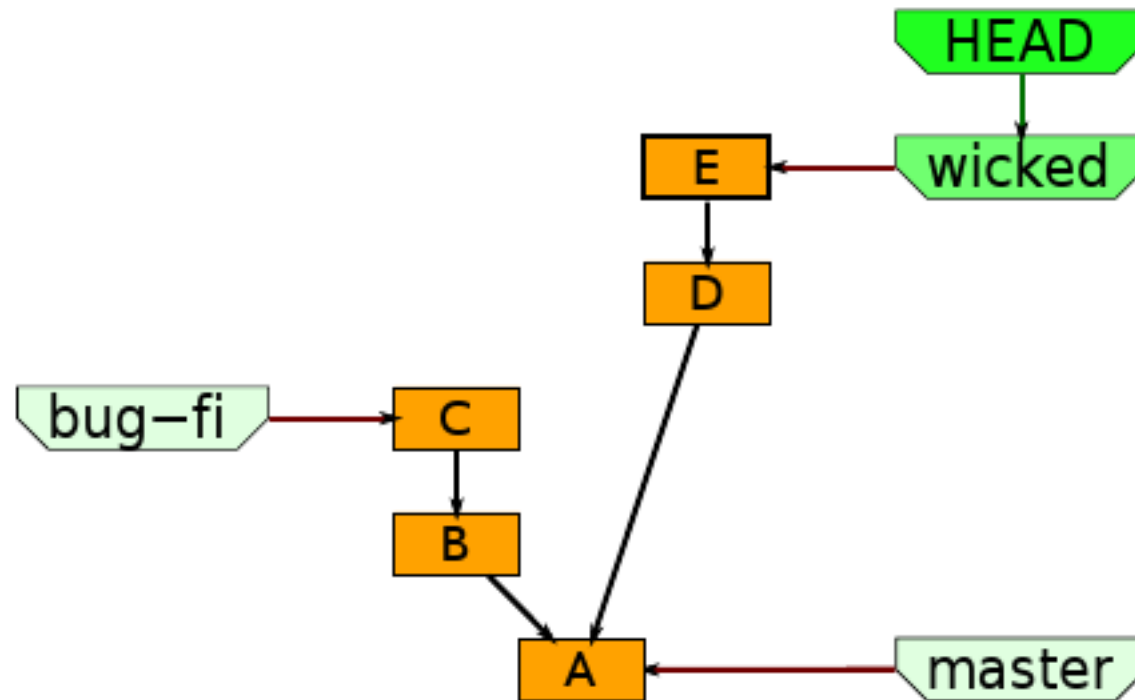
```
git checkout -b wicked  
master
```



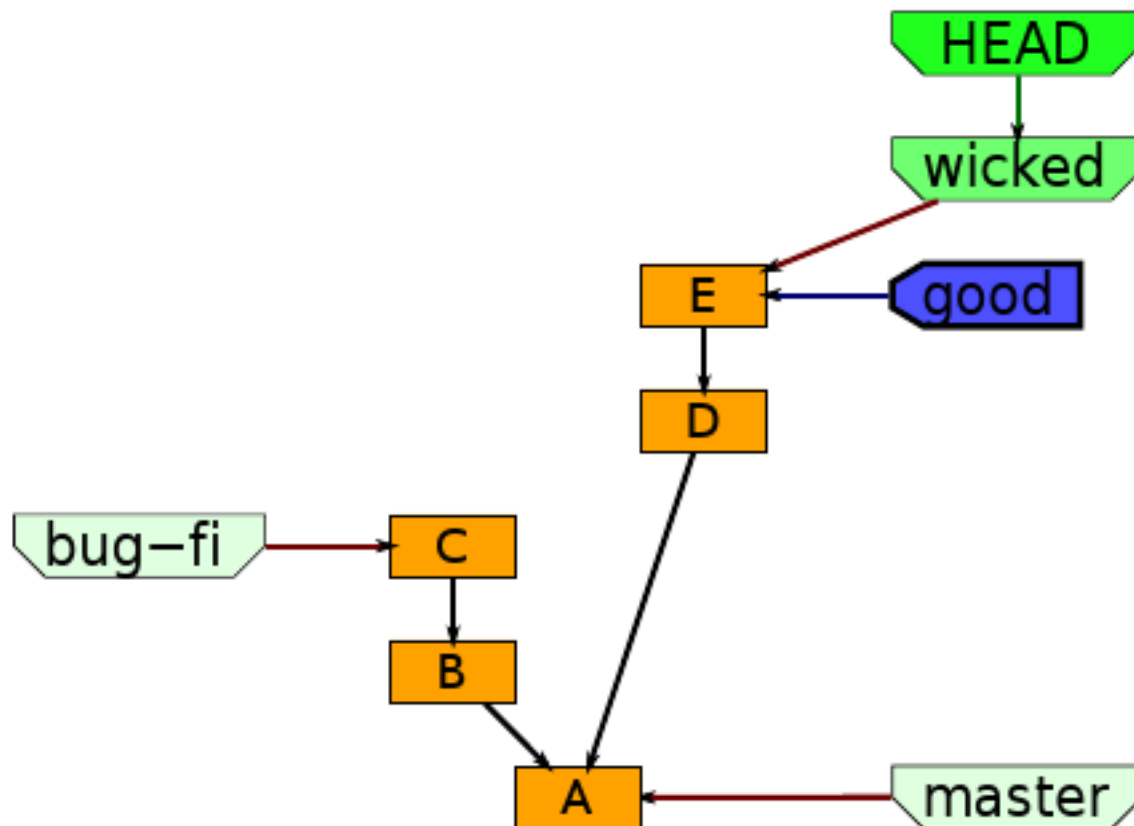

```
git commit -a -m "D"
```



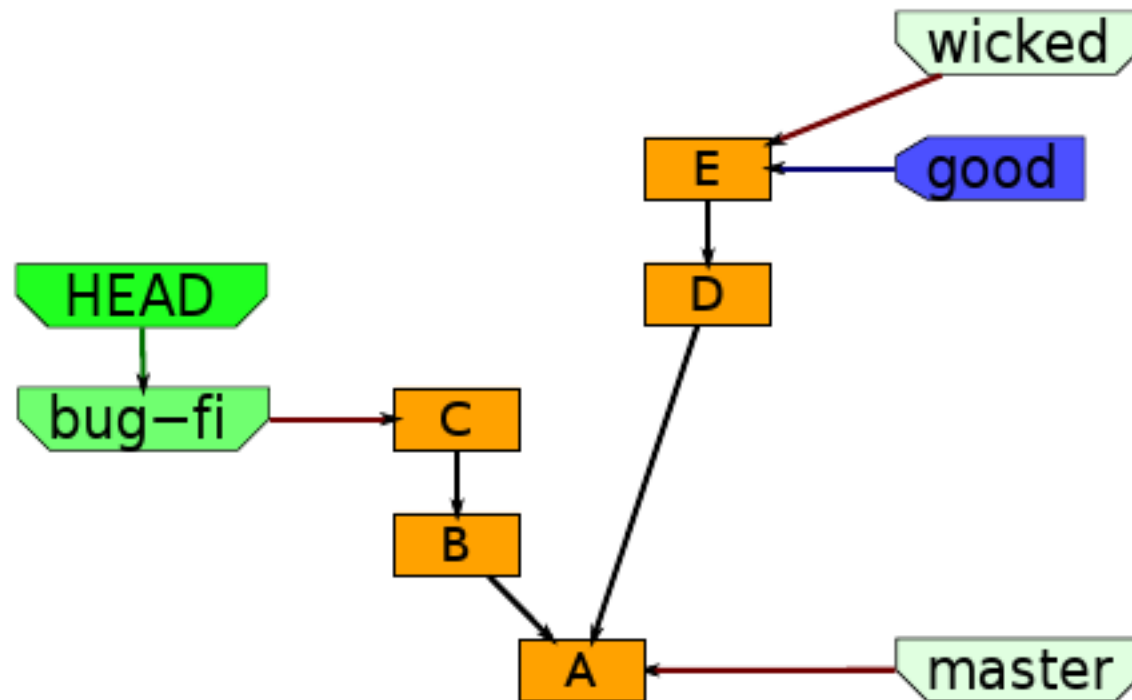
```
git commit -a -m "E"
```



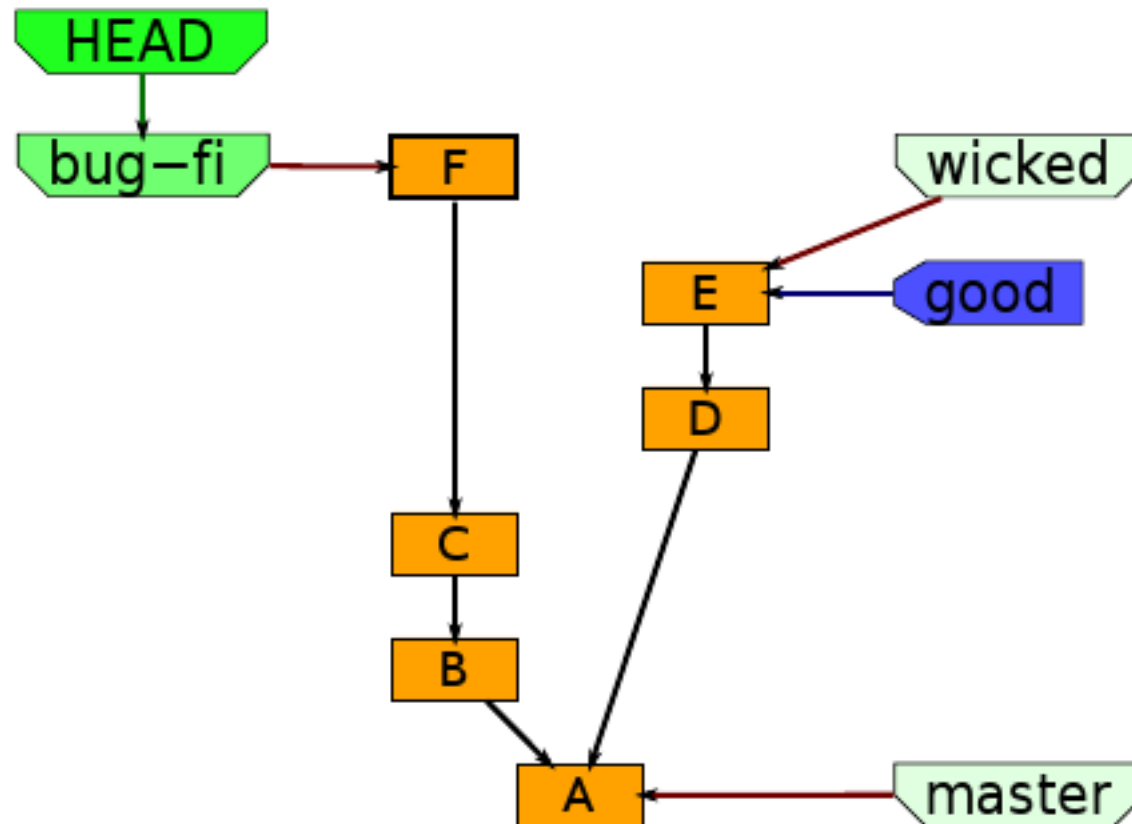
git tag -a -m "works" good



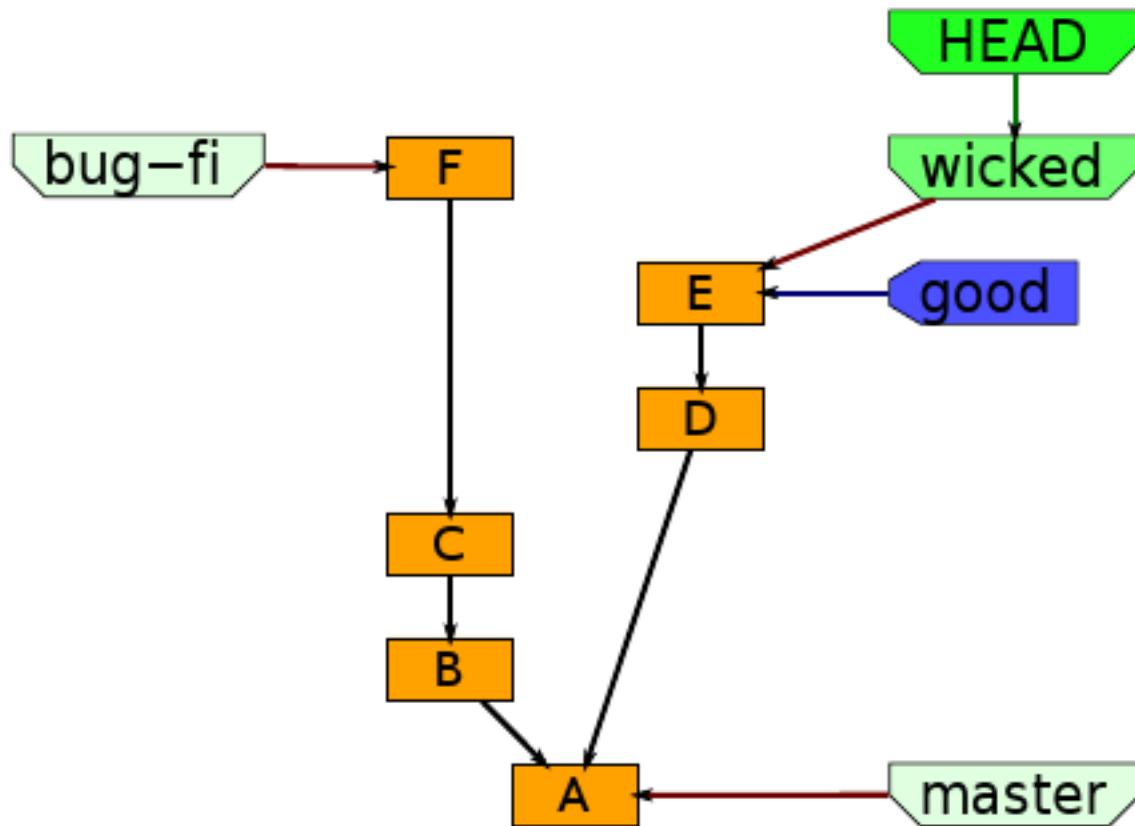
git check-out bug-fix



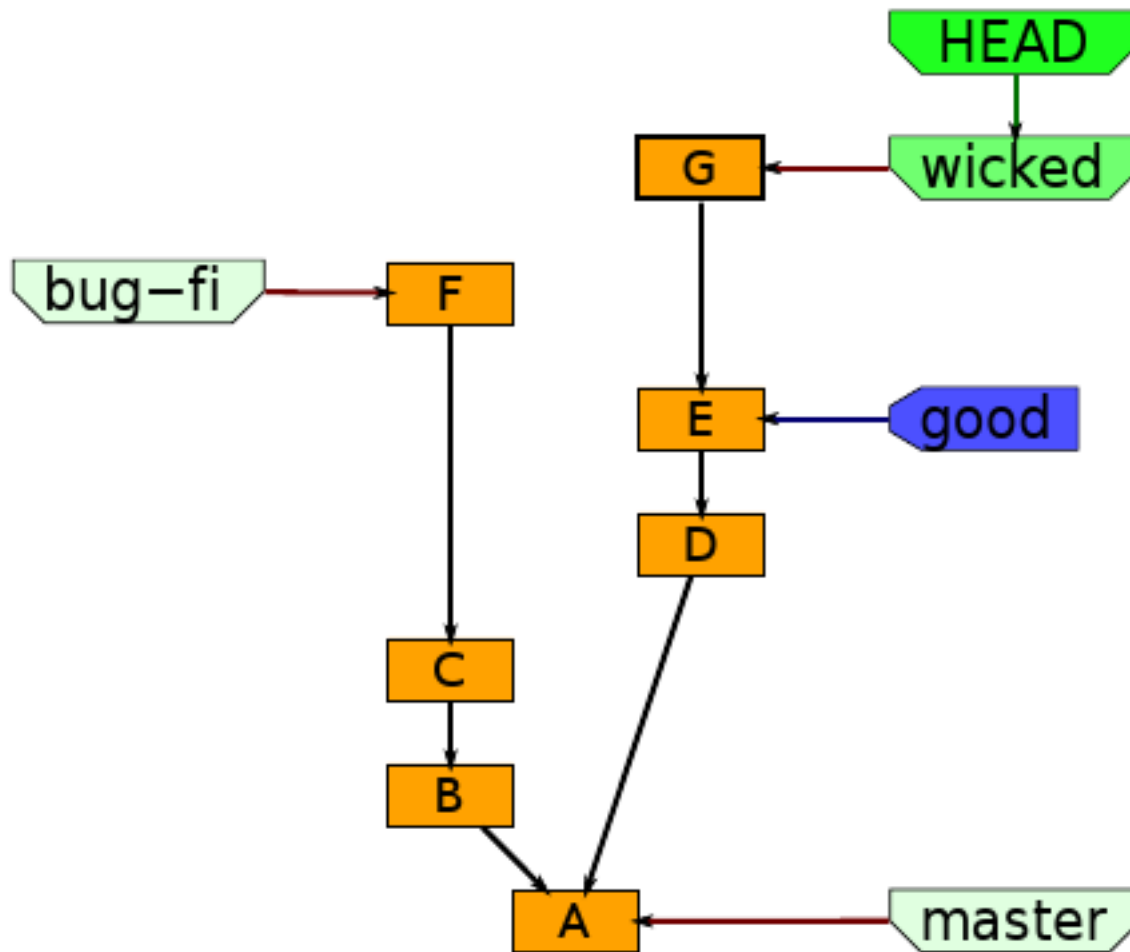
```
git commit -a -m "F"
```



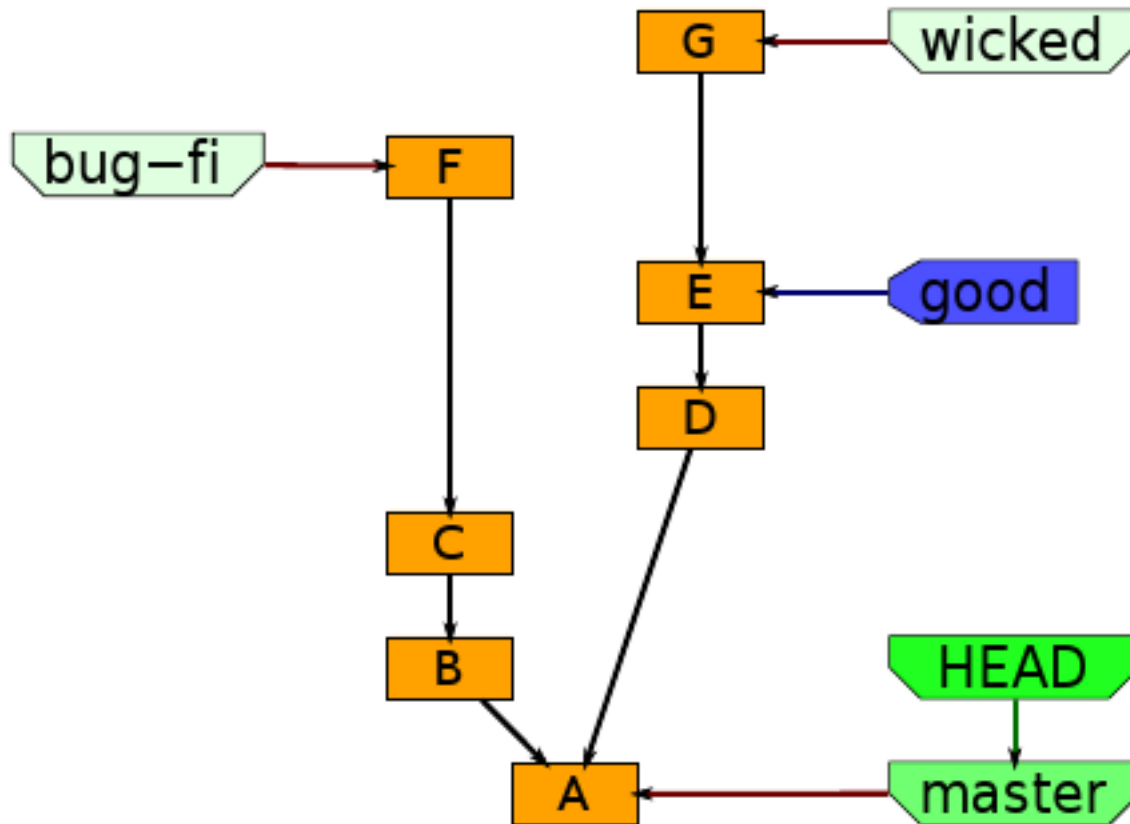
git checkout wicked



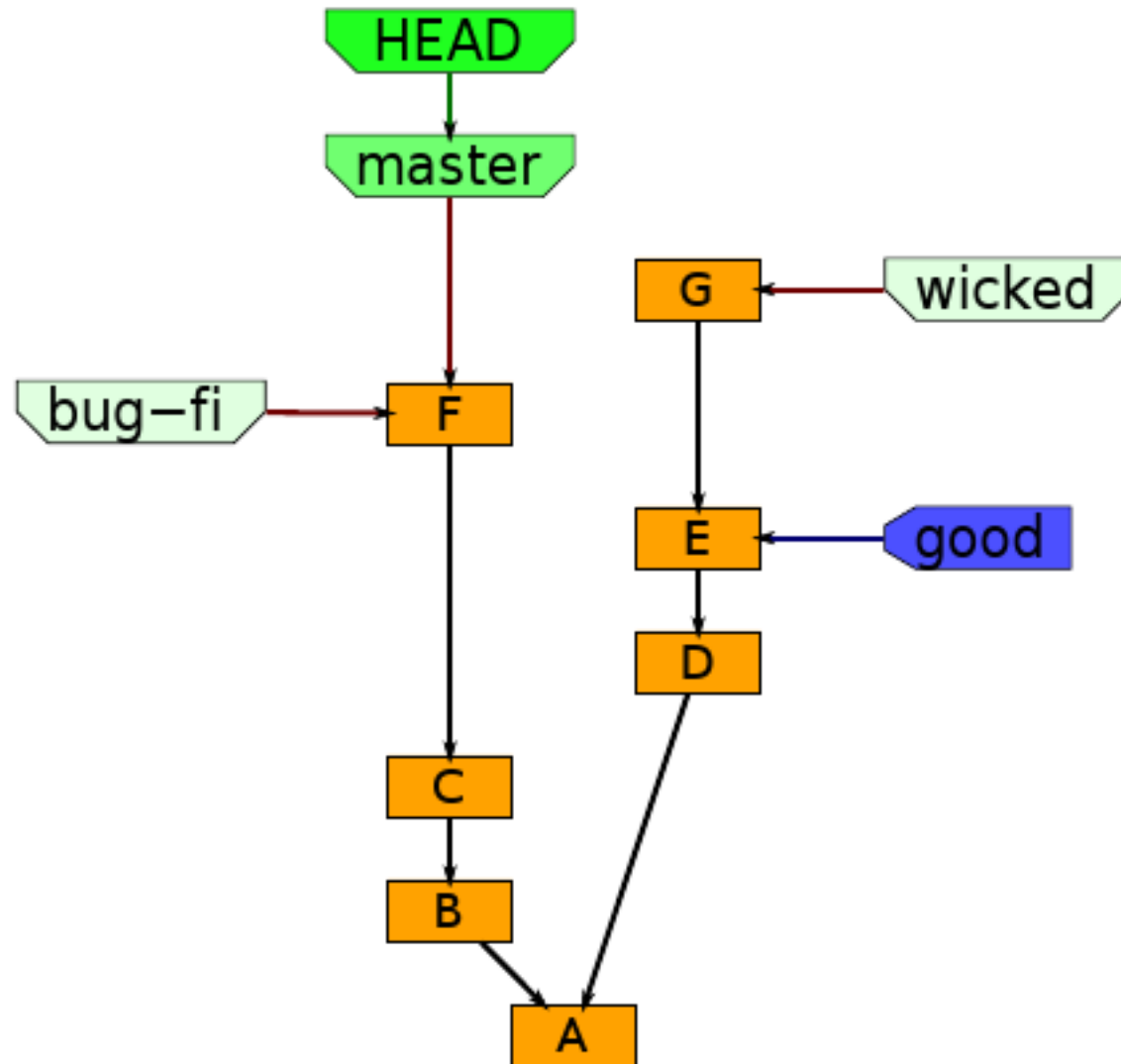
```
git commit -a -m "G"
```



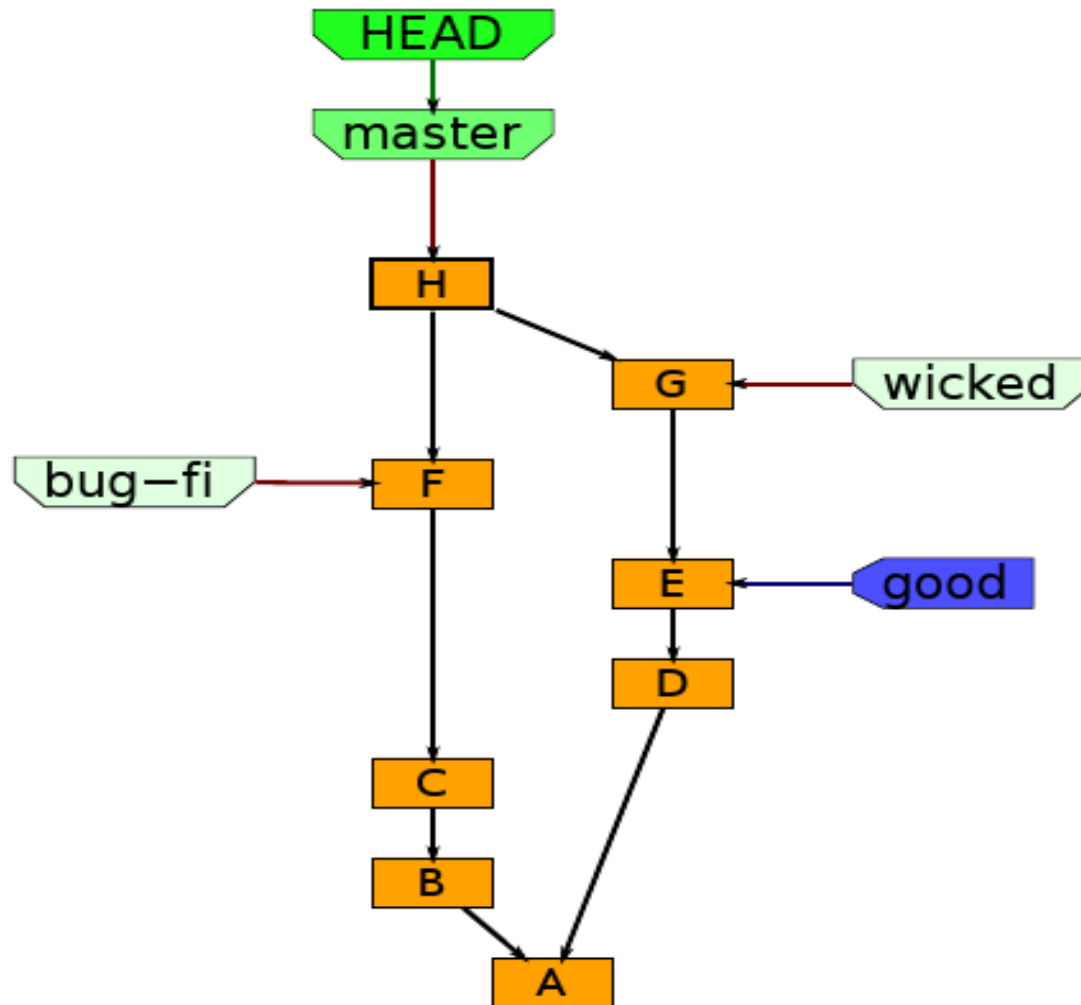
git checkout master



`git reset --hard bug-fix`



git merge wicked



git Tools

git log

git diff

git grep

git show

git status

git help [command]



Thanks!

Especially to Bart Trojanowski
for the DAG graphs