

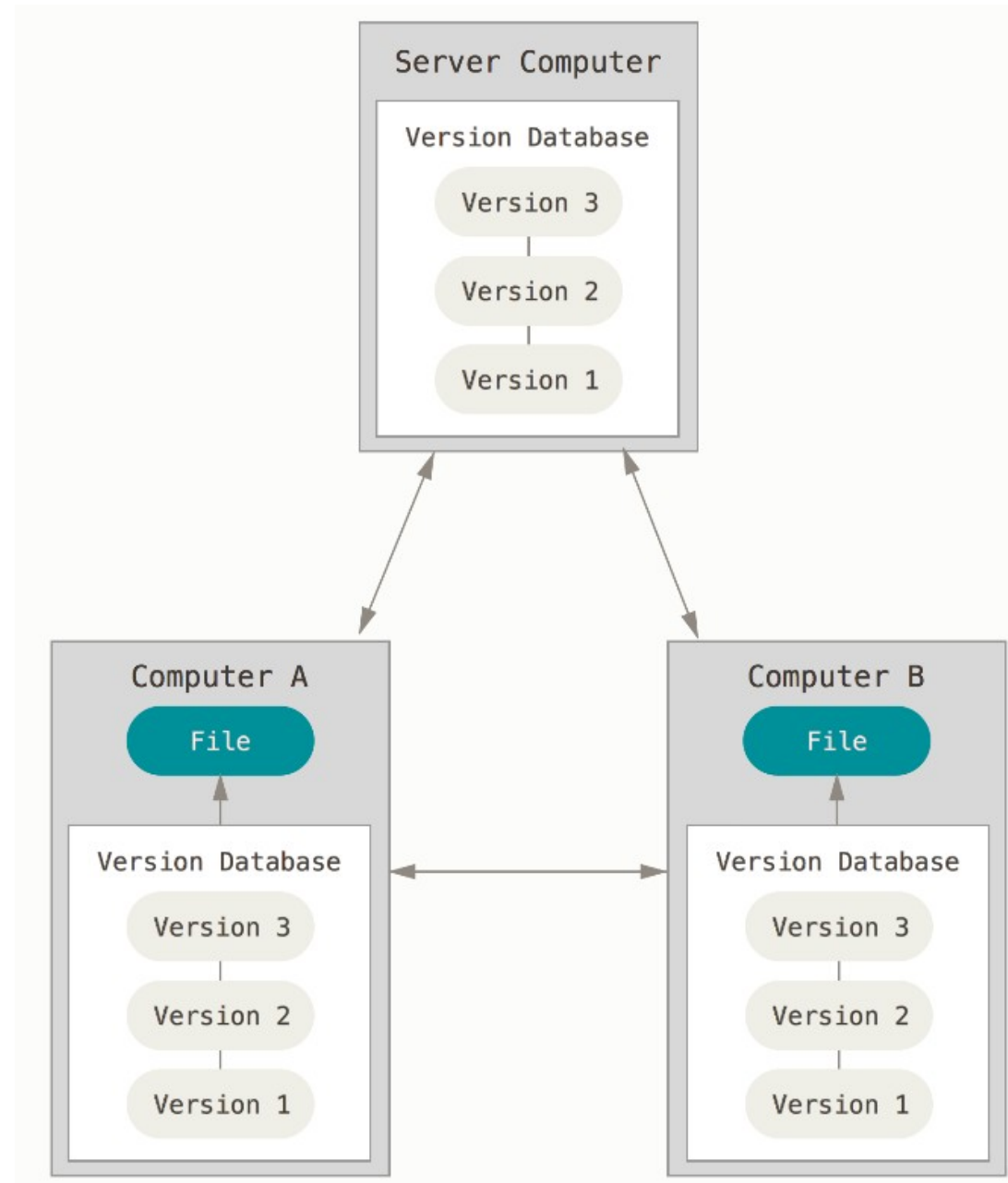
Git and GitHub

Mark Scott for the vPRISM collaboration
vPRISM analysis meeting
17th March 2015 - IPMU

I am not an expert – please shout out if I say something wrong, if you have a question, or if there's a better way to do something

What is Git?

- Version control software
- DVCS – Distributed Version Control System
 - CVS is a Centralised version control system
 - Each copy of the repository has the complete history of the project and fully mirrors everything in the original version
 - No single repository is the 'official' repository
 - Can have multiple remote repositories - allows easy collaboration



What is GitHub?

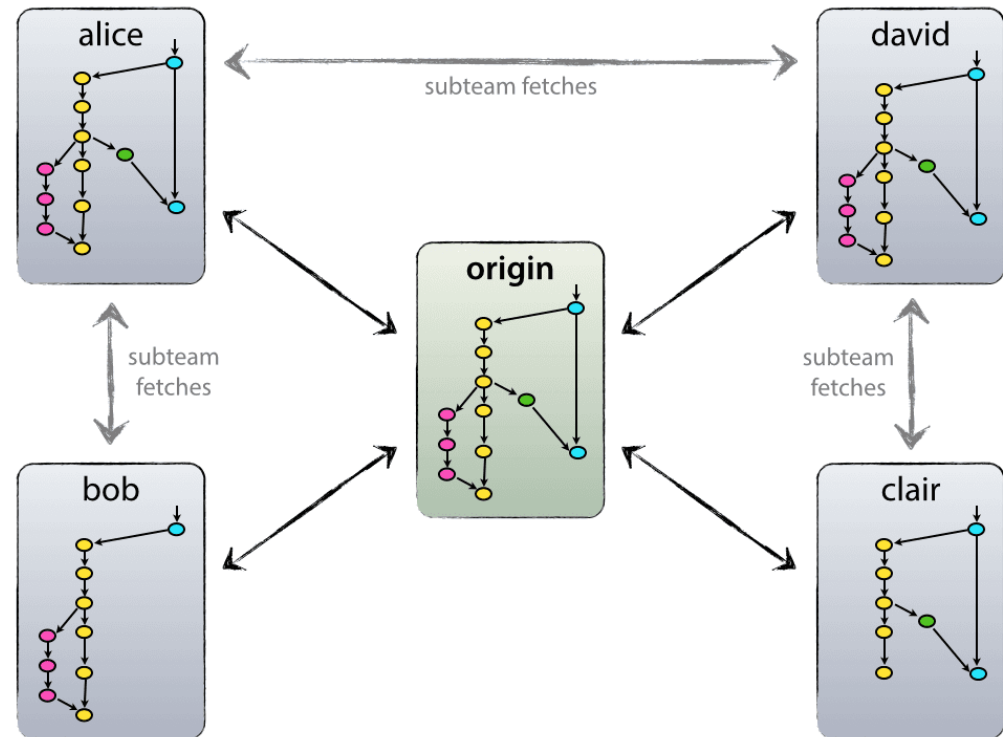
- Web-based Git repository host
 - Graphical interface to Git repositories
 - Allows easy bug tracking, documentation and feature requests
 - Introduces a more 'official' version of the repository
 - Allows more centralised control of repositories
- Used by WCSim and Hyper-K – we will follow their work flow system
- <https://github.com/nuPRISM> - this is the nuPRISM organisation
- You should create an account and 'fork' each repository from the nuPRISM organisation to your personal account
- A 'fork' is the GitHub name for the 'git clone' command – it creates a copy of the repository in your account



Git branches

- In git, when you work on some new code you make a new branch
 - A cheap operation, unlike CVS
 - Lets you work separately on multiple features at once – each on a different branch

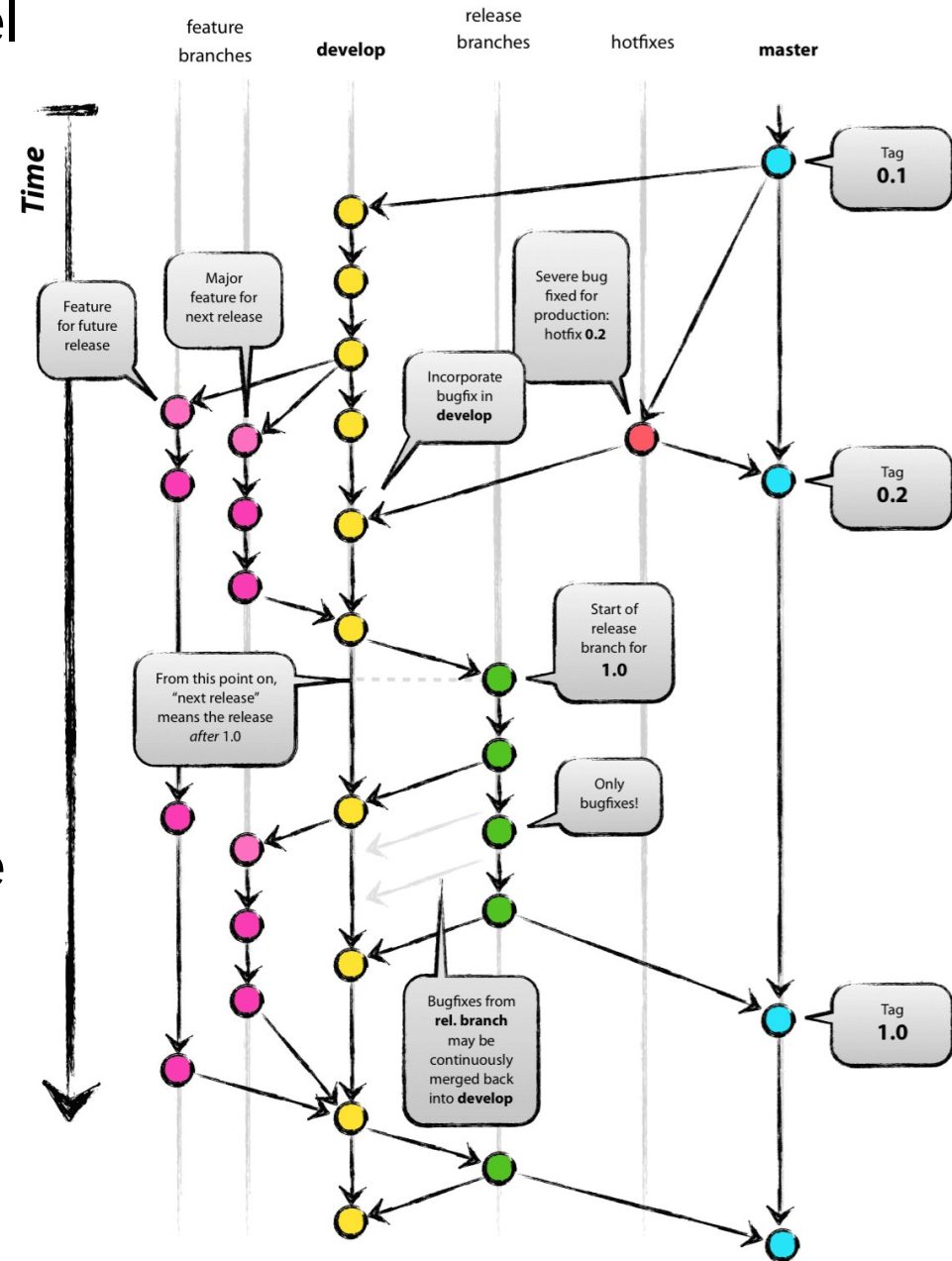
- Allows collaborative working
 - You can commit code to your branch that stops the software compiling without affecting anyone else
 - Many people can work on a single branch



- You should **always** submit new code from a new branch

Using Git(Hub)

- We are following **this** branching model
 - Used by WCSim
 - According to the internet it works very well...
- Two main branches:
 - master** – the production branch, should only be modified when we perform a production release
 - master branch should always compile and work!
 - develop** – the branch which is the basis for any software development
 - develop is not guaranteed to compile or work – similar to the HEAD version in CVS



Developing a feature

- First, clone your GitHub repository to your local machine (should have been done yesterday by the nuPRISM installation script)

```
git clone git@github.com:msscott201/WCSim.git
```

- Next, check which branch you are on

`git branch` - this lists all the branches in your local repository, and the * indicates which branch you are on

```
markscott@TRIUMF:~/T2K/nuPRISM_Software/Analysis$ git branch
* develop
```

- Now, create a new branch, called myFeature

```
git checkout -b feature/myFeature develop
```

- 'git checkout -b' will switch to a branch, and create a new branch if it doesn't exist
- The next argument is the branch name
- The final argument is the branch you want your new branch to be created from

Developing a feature - 2

- Type 'git branch' again – we're now on the feature/myFeature branch

```
markscott@TRIUMF:~/T2K/nuPRISM_Software/Analysis$ git branch
develop
* feature/myFeature
```

- Now, make any code changes you want – commit often and with informative commit messages (these are all stored in the repo history)

```
git commit filename.cxx
```

- There are other ways of committing things, but this is fairly foolproof
- Try the 'git status' command at some point

```
# On branch feature/myFeature
# 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:   Source_At_Start_nuPRISM.sh
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       Prob3++/BargerPropagator.o
#       Prob3++/EarthDensity.o
```

- Lists all modified files and files not being tracked by git – use 'git add file.cxx' to add a new file to the repository

Remote repositories

- We've made the changes we want, and tested that everything works
- Now, need to 'push' these changes to your GitHub repository. First, check which remote repositories you have set up

```
git remote -v  
markscott@TRIUMF:~/T2K/nuPRISM_Software/Analysis$ git remote -v  
origin https://github.com/msscott201/Analysis.git (fetch)  
origin https://github.com/msscott201/Analysis.git (push)
```

- Remote repositories are repositories that you can 'push' updates to or 'pull' updates from
 - One of the good things about git!
- Collaborate with Mark Hartz on something...

```
git remote add hartz https://github.com/markhartz/Analysis.git
```

- Adds a repo call 'hartz' that points to Mark's Analysis repository

```
markscott@TRIUMF:~/T2K/nuPRISM_Software/Analysis$ git remote -v  
mhartz_repo https://github.com/markhartz/Analysis.git (fetch)  
mhartz_repo https://github.com/markhartz/Analysis.git (push)  
origin https://github.com/msscott201/Analysis.git (fetch)  
origin https://github.com/msscott201/Analysis.git (push)
```

- By default, when cloning a repo, a remote repository called origin is added, pointing to the remote repo you cloned from

Remote repositories - 2

- So, I now have two remote repositories, my repo on GitHub and Mark's
- If we're both working on one thing, if Mark makes a change to his repo I can 'pull' that change over to my local code

```
git fetch mhartz
```

```
markscott@TRIUMF:~/T2K/nuPRISM_Software/Analysis$ git fetch mhartz_repo
Username for 'https://github.com': mscott201
Password for 'https://mscott201@github.com':
remote: Counting objects: 9, done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 3), reused 0 (delta 0), pack-reused 2
Unpacking objects: 100% (9/9), done.
From https://github.com/markhartz/Analysis
* [new branch]      develop    -> mhartz_repo/develop
* [new branch]      master     -> mhartz_repo/master
```

- 'git fetch' create a local copy of the remote repository, stored under repo_name/branch_name
- Can then merge changes into my current branch

```
git merge mhartz/develop – it always merges into your current branch
```

```
markscott@TRIUMF:~/T2K/nuPRISM_Software/Analysis$ git branch
  develop
* feature/myFeature
markscott@TRIUMF:~/T2K/nuPRISM_Software/Analysis$ git merge mhartz_repo/develop
Updating a58123f..da29ef8
Fast-forward
 FluxFits/fit_gaussian.cc | 228 ++++++
 1 file changed, 228 insertions(+)
 create mode 100644 FluxFits/fit_gaussian.cc
```

Pushing a feature

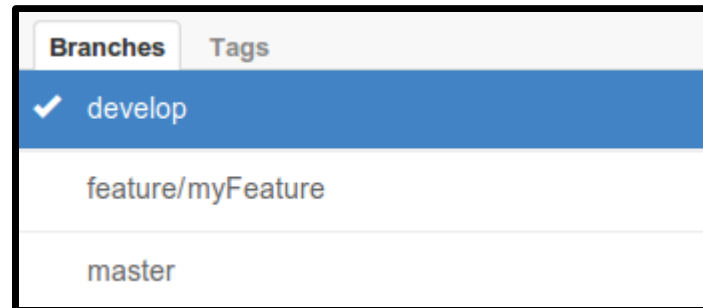
- Want to 'push' my last changes up to my GitHub repository so that Mark can get them

`git status` – check everything I want to 'push' has been committed

`git push origin feature/myFeature`

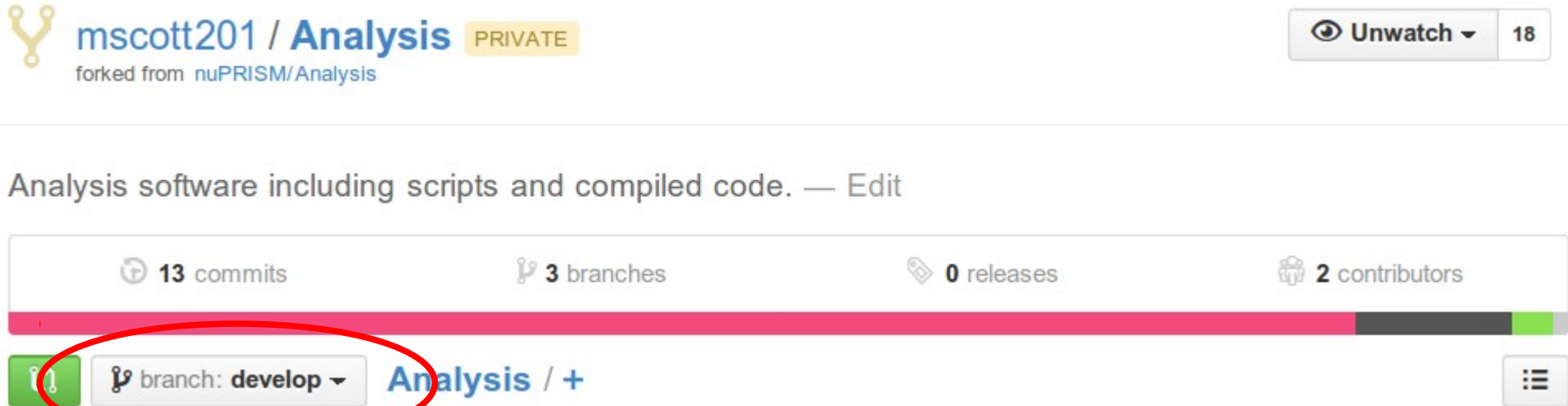
```
markscott@TRIUMF:~/T2K/nuPRISM_Software/Analysis$ git push origin feature/myFeature
Username for 'https://github.com': mscott201
Password for 'https://mscott201@github.com':
To https://github.com/mscott201/Analysis.git
 * [new branch]      feature/myFeature -> feature/myFeature
```

- You now have a new branch in your GitHub account called feature/myFeature



- Carry on committing, pushing and pulling code between people's personal repositories until the myFeature branch is finished and validated

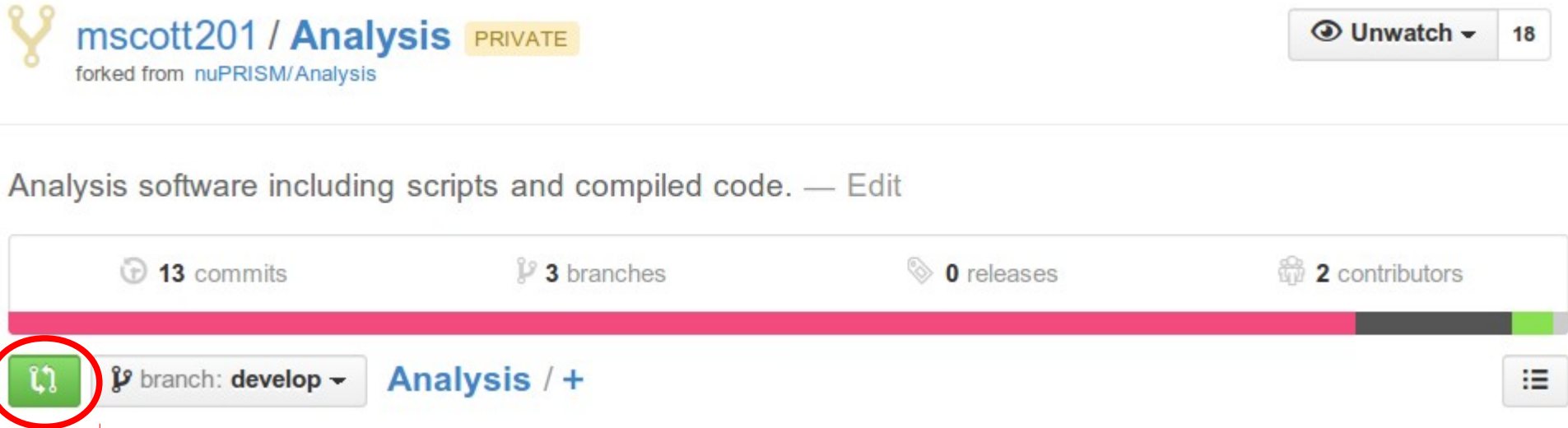
- When a feature is complete, you can use GitHub to submit a pull request to the nuPRISM organisation repository



The screenshot shows a GitHub repository page for 'msscott201 / Analysis' (PRIVATE), forked from 'nuPRISM/Analysis'. The repository has 13 commits, 3 branches, 0 releases, and 2 contributors. A red circle highlights the 'branch: develop' dropdown menu, which is currently selected. The repository name 'Analysis / +' is also visible.

- First, select your feature branch

- When a feature is complete, you can use GitHub to submit a pull request to the nuPRISM organisation repository



The screenshot shows a GitHub repository page for 'msscott201 / Analysis', which is a private fork of 'nuPRISM/Analysis'. The repository statistics are: 13 commits, 3 branches, 0 releases, and 2 contributors. A horizontal bar at the bottom of the repository shows a green segment on the left, a dark grey segment in the middle, and a small green segment on the right. A green button with a pull request icon is circled in red. Below the button, the text 'branch: develop' is visible, followed by 'Analysis / +' and a menu icon.

- First, select your feature branch
- Then, click the green button on the left

- See a screen like this

nuPRISM / Analysis
PRIVATE

Unwatch ▾ 20

★ Star 0

Y Fork 12

base fork: nuPRISM/Analysis ▾

base: develop ▾

...

head fork: mscott201/Analysis ▾

compare: feature/myFeature ▾

↔

Create pull request

Discuss and review the changes in this comparison with others.

?

↻

 1 commit

📄

 1 file changed

💬

 0 commit comments

👤

 1 contributor

📅

 Commits on Mar 16, 2015

↻

👤

 mscott201

Demonstration commit

826054a

📄

 Showing 1 changed file with 0 additions and 1 deletion.

Unified

Split

1

■
■
■
■
■
■

nuPRISM_Install.sh

View

✚		@@ -2,7 +2,6 @@
2	2	# Script to checkout a HyperK release from git. The script
3	3	# reads a config file containing the packages and revisions
4	4	# and checks out those repositories
5	-	-#
6	5	
7	6	nuPrism_config="nuPRISM_release.cfg"
8	7	
✚		

4

- See a screen like this

nuPRISM / Analysis PRIVATE

The base branch, which would pull our changes (the nuPRISM/develop branch in almost all cases)

base fork: nuPRISM/Analysis base: develop ... head fork: mscott201/Analysis compare: feature/myFeature

Create pull request Discuss and review the changes in this comparison with others.

1 commit 1 file changed 0 commit comments 1 contributor

Commits on Mar 16, 2015

mscott201 Demonstration commit 826054a

Showing 1 changed file with 0 additions and 1 deletion. Unified Split

1 nuPRISM_Install.sh View

		@@ -2,7 +2,6 @@
2	2	# Script to checkout a HyperK release from git. The script
3	3	# reads a config file containing the packages and revisions
4	4	# and checks out those repositories
5	5	-#
6	6	nuPrism_config="nuPRISM_release.cfg"
7	7	

The repository and branch we are sending the request from – it has our new feature

Unwatch 20 Star 0 Fork 12

base fork: nuPRISM/Analysis base: develop ... **head fork: mscott201/Analysis compare: feature/myFeature**

Create pull request Discuss and review the changes in this comparison with others.

1 commit 1 file changed 0 commit comments 1 contributor

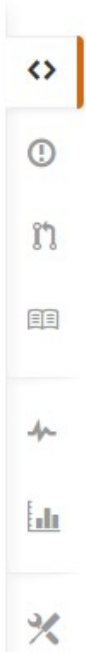
Commits on Mar 16, 2015
 mscott201 Demonstration commit 826054a

Showing 1 changed file with 0 additions and 1 deletion. Unified Split

1 nuPRISM_Install.sh View

```

@@ -2,7 +2,6 @@
2 # Script to checkout a HyperK release from git. The script
3 # reads a config file containing the packages and revisions
4 # and checks out those repositories
5 -#
6
7 nuPrism_config="nuPRISM_release.cfg"
8
    
```



- See a screen like this

nuPRISM / Analysis PRIVATE Unwatch 20 Star 0 Fork 12

base fork: nuPRISM/Analysis base: develop head fork

A summary of the changes and commit messages

Create pull request Discuss and review the changes in this comparison with others.

1 commit 1 file changed 0 commit comments 1 contributor

Commits on Mar 16, 2015

mscott201 Demonstration commit 826054a

Showing 1 changed file with 0 additions and 1 deletion. Unified Split

1 nuPRISM_Install.sh View

		@@ -2,7 +2,6 @@
2	2	# Script to checkout a HyperK release from git. The script
3	3	# reads a config file containing the packages and revisions
4	4	# and checks out those repositories
5		-#
6	5	
7	6	nuPrism_config="nuPRISM_release.cfg"
8	7	

- See a screen like this

 nuPRISM / Analysis PRIVATE Unwatch 20 Star 0 Fork 12

base fork: nuPRISM/Analysis base: develop ... head fork: mscott201/Analysis compare: feature/myFeature

[Create pull request](#) Discuss and review the changes in this comparison with others.

1 commit

1 file changed

0 commit comments

1 contributor

Commits on Mar 16, 2015

mscott201

Demonstration commit

826054a

The actual changes to the code

Showing 1 changed file with 0 additions and 1 deletion.

Unified

Split

1 nuPRISM_Install.sh

View

```
@@ -2,7 +2,6 @@
2      2      # Script to checkout a HyperK release from git. The script
3      3      # reads a config file containing the packages and revisions
4      4      # and checks out those repositories
5      5      -#
6      6      nuPrism_config="nuPRISM_release.cfg"
7      7
```

- See a screen like this

nuPRISM / Analysis PRIVATE

 Unwatch 20
 Star 0
 Fork 12

base fork: nuPRISM/Analysis base: develop ... head fork: mscott201/Analysis compare: feature/myFeature

Create pull request
Discuss
The button to submit the request

1 commit
 1 file changed
 0 commit comments
 1 contributor

Commits on Mar 16, 2015

mscott201 Demonstration commit 826054a

Showing 1 changed file with 0 additions and 1 deletion.

 Unified Split

1 ■ nuPRISM_Install.sh View

⌕	@@	-2,7 +2,6 @@
2	2	# Script to checkout a HyperK release from git. The script
3	3	# reads a config file containing the packages and revisions
4	4	# and checks out those repositories
5	-	#
6	5	
7	6	nuPrism_config="nuPRISM_release.cfg"
8	7	
⌕		

- Submitted request sent to the nuPRISM repository
- Request appears, detail changes (very similar to last slide)
- Allows users to comment on changes and suggest improvements etc.
- See here for an example - <https://github.com/nuPRISM/WCSim/pull/1>

- A repository manager will decide if the pull request should be merged or not
 - Will iterate with comments and might ask you to fix any conflicts that appear
 - **Any change you make to your feature/myFeature branch will also be included in the pull request** – if you want to do more work start a new branch
- Please do not push directly to the nuPRISM repository
- Please do not merge your own pull request without discussing it first

- Submitted request sent to the nuPRISM repository
- Request appears, detail changes (very similar to last slide)
- Allows users to comment on changes and suggest improvements etc.
- See here for an example - <https://github.com/nuPRISM/WCSim/pull/1>

- A repository manager will decide if the pull request should be merged or not
 - Will iterate with comments and might ask you to fix any conflicts that appear
 - **Any change you make to your feature/myFeature branch will also be included in the pull request** – if you want to do more work start a new branch
- Please do not push directly to the nuPRISM repository
- Please do not merge your own pull request without discussing it first

- Your pull request was accepted, and so your change is now in the develop branch of the nuPRISM repo – need to do a cleanup
- First, delete the old feature branch on your local machine

`git checkout develop` – move to the **develop** branch

`git branch -d feature/myFeature` – delete the old branch

```
markscott@TRIUMF:~/T2K/nuPRISM_Software/Analysis$ git checkout develop
M      Source_At_Start_nuPRISM.sh
Switched to branch 'develop'
markscott@TRIUMF:~/T2K/nuPRISM_Software/Analysis$ git branch -d feature/myFeature
error: The branch 'feature/myFeature' is not fully merged.
If you are sure you want to delete it, run 'git branch -D feature/myFeature'.
markscott@TRIUMF:~/T2K/nuPRISM_Software/Analysis$ git branch -D feature/myFeature
Deleted branch feature/myFeature (was 826054a).
```

- Notice the first warning – I had not merged all my changes from that branch onto another, so I can't just delete it by accident

`git push origin :feature/myFeature` – delete the feature/myFeature branch from the origin repository

```
markscott@TRIUMF:~/T2K/nuPRISM_Software/Analysis$ git push origin :feature/myFeature
Username for 'https://github.com': mscott201
Password for 'https://mscott201@github.com':
To https://github.com/mscott201/Analysis.git
- [deleted]          feature/myFeature
```

- Now want to merge back this new feature into my local and GitHub repo's
- First, add the nuPRISM repo as a remote repository called 'upstream'

```
git remote add upstream https://github.com/nuPRISM/Analysis.git
```

- Make sure you are on your **develop** branch

```
git fetch upstream
```

```
git merge upstream/develop
```

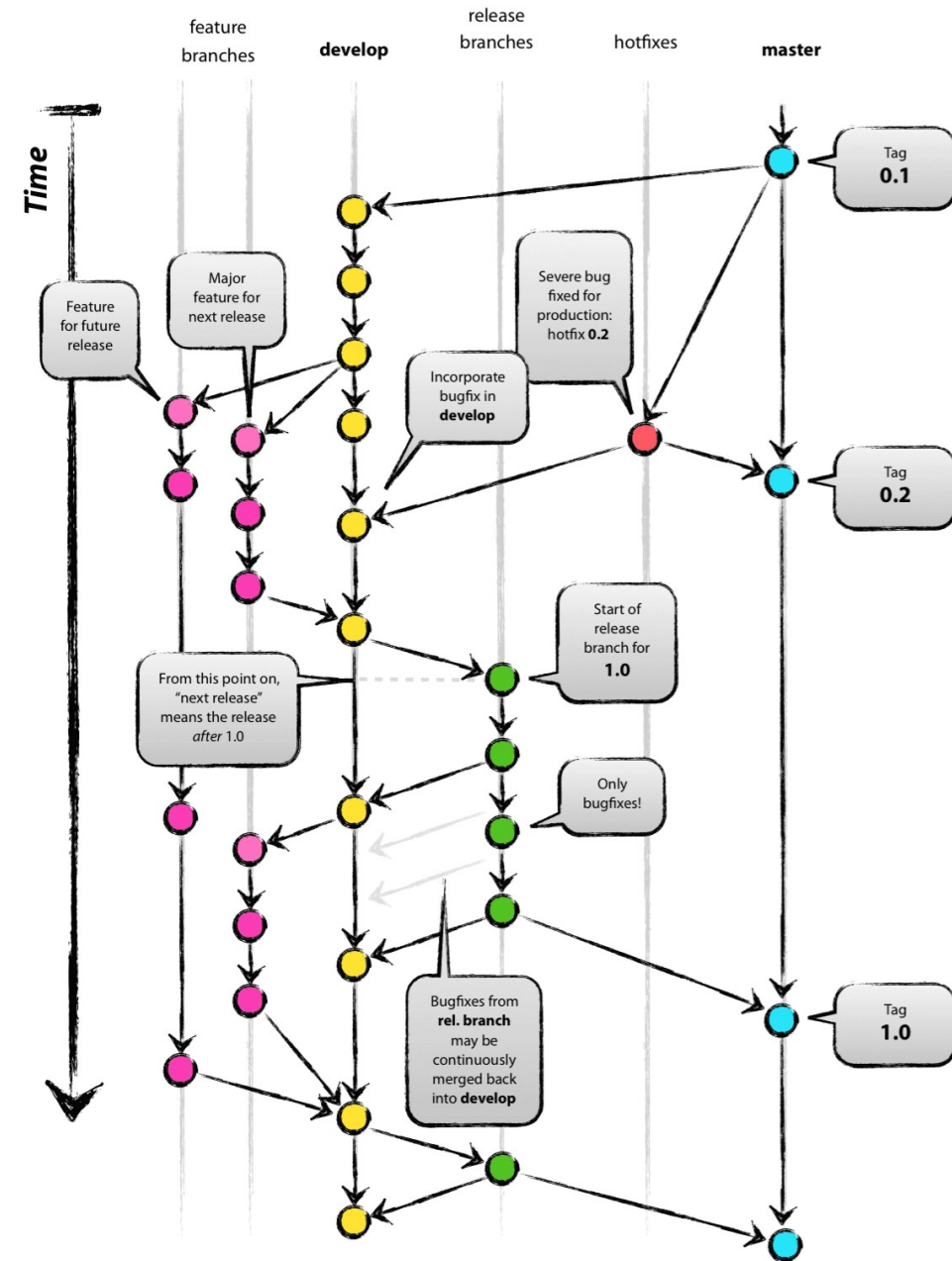
- Now your develop branch is level with the nuPRISM develop branch

```
git push origin develop
```

- Now your GitHub repo matches the nuPRISM develop branch as well

- Make a new branch and start the process over again!

- **Feature** branches:
 - Must be branched from the **develop** branch
 - Must merge (pull request) back into the **develop** branch
- **Release** branch is used to release software
 - Branches from **develop**
 - Merges into **develop** and **master**
- **Hotfix** branch – maybe not needed?
 - Branches from **master**
 - Merges into **master** and **develop**
- Any time a change merges into the **master** branch this is a software release – tag a version



- Just a quick overview to help people get started
- Lots of help available online
- Strongly recommend you read these pages:
 - <http://git-scm.com/book/en/v2>
 - <http://nvie.com/posts/a-successful-git-branching-model/>
- Some useful commands:
 - 'git status' – shows any changes and what is being tracked by git
 - 'git branch -a' – shows all branches available, including remote ones
 - 'git remote -v' – gives information on all the remote repos you have
 - 'git checkout' – switch between different branches
 - 'git diff file.cxx' – shows the difference between your current file and whatever was last committed to git (the local repository)