Git-Flow and "Add a Test" Activity
JEDI Academy: 4-7 June, 2018, Boulder, CO

Step 1) Build ufo-bundle with feature/ufo-tutorial branch of ufo

(Most of this should be review):

- Clone ufo-bundle in some suitable source directory (git clone https://github.com/JCSDA/ufo-bundle.git)
- Edit CMakeLists.txt to select feature/ufo-tutorial repo in place of ufo

  ecbuild_bundle(PROJECT ufo GIT "https://github.com/JCSDA/**ufo-training.git**" BRANCH **feature/ufo-tutorial** UPDATE)

- Also comment out builds of eckit and fckit if you are in the container (optional)
- Clean (rm -rf *) your build directory and run ecbuild (on ufo-bundle)
- Make sure you have the latest versions (not necessary if your ufo bundle is new, but it's good to get in the habit):

  ```
  > make update
  ```
- Compile

  ```
  > make -j4
  ```
- Test

  ```
  > ctest
  ```

Step 2) install git flow

In container:

no action needed: move to Step 3

On Mac:

```
> brew install git-flow-avh
```

On ubuntu:

```
> sudo apt-get install git-flow
```

otherwise google git flow avh

If you don't have root privileges (and you're not in the container), you can skip this step

<u>Step 3) initialize git flow in the ufo-training repo</u>

Go to ufo-bundle/ufo-training source directory and initialize git-flow

manual initialization:

```
> git flow init
```

Read the prompts with care, but just press return to select all the defaults

Shortcut (for future reference)

```
> git flow init -d
```

Or, to force defaults:

```
> git flow init -f -d
```

<u>4) Create a feature branch</u>

```
> git flow feature start academy-<initials>
```

Or, if you cannot use git flow, this is equivalent to

```
> git branch feature/academy-<initials>
> git checkout feature/academy-<initials>
```

<u>5) Create a new config file</u>

Normally, you would probably add a new feature in your feature branch and then add a test to test it. But today, we'll pretend that you already added your feature and now you want to test it.

Go to the ufo/test/testinput directory (in your source directory)

Copy the file ufotest.json to a new file with a name of your choice (keep the .json)

Delete or change the Locations entry in the config file (a trivial example of a more significant change you may wish to make in the configuration - a real new test would require you to also change a test result, such as a norm - but no need to do that now)

<u>6) Add a new test that uses your new config file</u>

Edit ufo/test/CMakeLIsts.txt (in your source directory)

- Append your new config file to the ufo_test_input list
  - ***(near the top - look for where ufotest.json is)***
- Add a test with ecbuild_add_test()
  - ***(use the entry for test_ufo_geovals as a template - copy, paste, give your test a name, and replace the config file with your new one)***

7) Build and test ufo-bundle using your modified code

Edit ufo-bundle/CMakeLIsts.txt (in your source directory) - tell ecbuild to build with your local copy instead of pulling from GitHub:

```
#ecbuild_bundle( PROJECT ufo   GIT “https://github.com/JCSDA/ufo-training.git"  BRANCH feature/ufo-tutorial UPDATE )
ecbuild_bundle( PROJECT ufo SOURCE “<src-directory>/ufo-training )
```

Go to your build directory and re-compile

- ***clean the directory and build it all again (to link the new config file)***

Run your new test

```
> ctest -R <yourtest>
```

Did it pass? Normally, you’d likely have to do more work to get your test to pass…

6) Commit and Publish your feature branch

Commit your changes to your local git feature branch

Cd to your ufo-training repo

```
> git add test/testinput/<yourconfig>.json
> git commit -a -m "<message>"
```

Publish your new feature branch to GitHub

```
> git flow feature publish academy-<initials>
```

- verify that the branches now exist on GitHub

If you don’t have git-flow, this is equivalent to:

```
> git push —set-upstream origin feature/academy-<initials>
```

7) terminating a feature branch

We will do this on Thursday afternoon - this is just to give you an idea of the full life cycle of a feature branch

- merge with current develop branch

```
> git checkout develop
> git pull
> git checkout feature/academy-<initials>
> git merge develop
> git push
```

Normally, you would proceed as follows **(today is not normal!)**

[do a pull request on GitHub - wait for review - remote branch should be terminated after merge]

First ensure that the remote branch is gone:

```
> git remote update -p
> git branch -a
```

Then manually delete your local branch **(do not do this now - you'll need it tomorrow!)**

```
> git branch -d feature/academy-<initials>
```

ZenHub Activity (if you have time)
JEDI Academy: 4-7 June, 2018, Boulder, CO

1) Sign up for a ZenHub account

- http://www.zenhub.com
- Sign up with GitHub account
- Install browser extension (Chrome or FireFox)

2) Access ZenHub board for JCSDA/ufo-training

- If not using extension, go through ZenHub page

3) Add an issue

- assignees
- Label
- difficulty

4) Add a checklist/sub-tasks to your issue

- with Markdown

5) Cards as Issues

- view issues in the GitHub "Issues" tab for the repo
- go to GitHub issues page to see all issues assigned to them

6) Prioritize Issues

- Among and within columns
- Collapse icebox

7) Define a Milestone (Sprint)

- Select and add issues
- Filter by Milestone
- collapse irrelevant pipelines

8) Review

- Move completed items to review/QA column
- Assign reviewers
- Comment on someone else’s issue