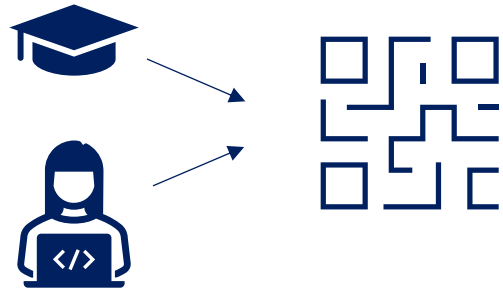


SVN To Git: Lessons Learnt From CABLE

Claire Carouge, Ian Harman, Jhan Srbinovsky, Paul Leopardi

Similarities and differences in code management and communities



Scientific codes with developments from academics and RSEs

JULES

CABLE



Community size



Defined workflow for release

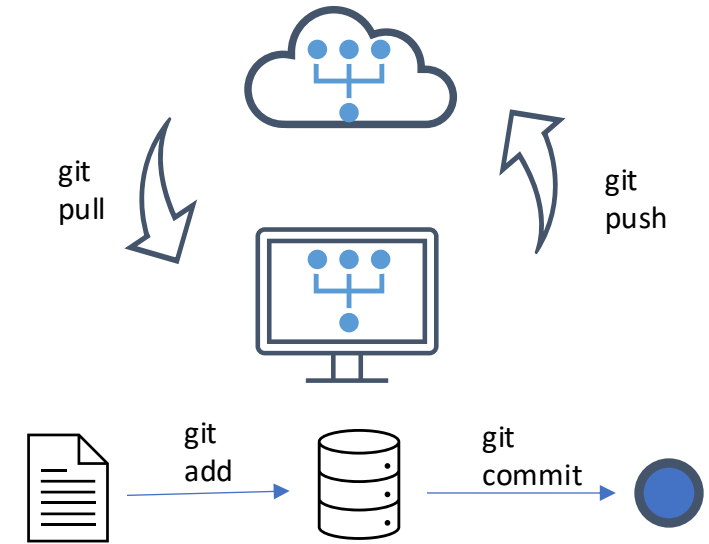


THE CHALLENGES

New tools = learning curve!

New concepts to learn:

- local repository
- 2-steps commits
- Local commits
- Push and pull for synchronization with remote repository
- GitHub interface



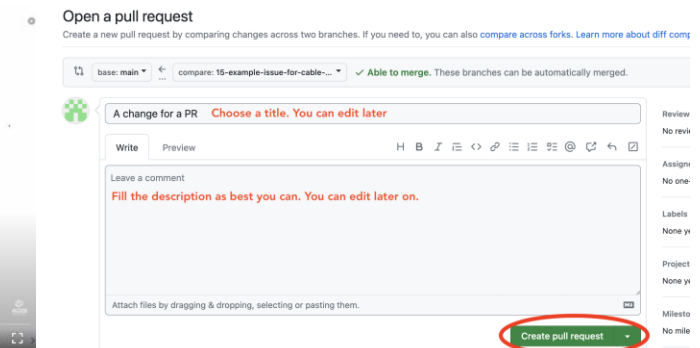
Training is essential

On-demand training material!

Community of practice or community help is also highly valuable.

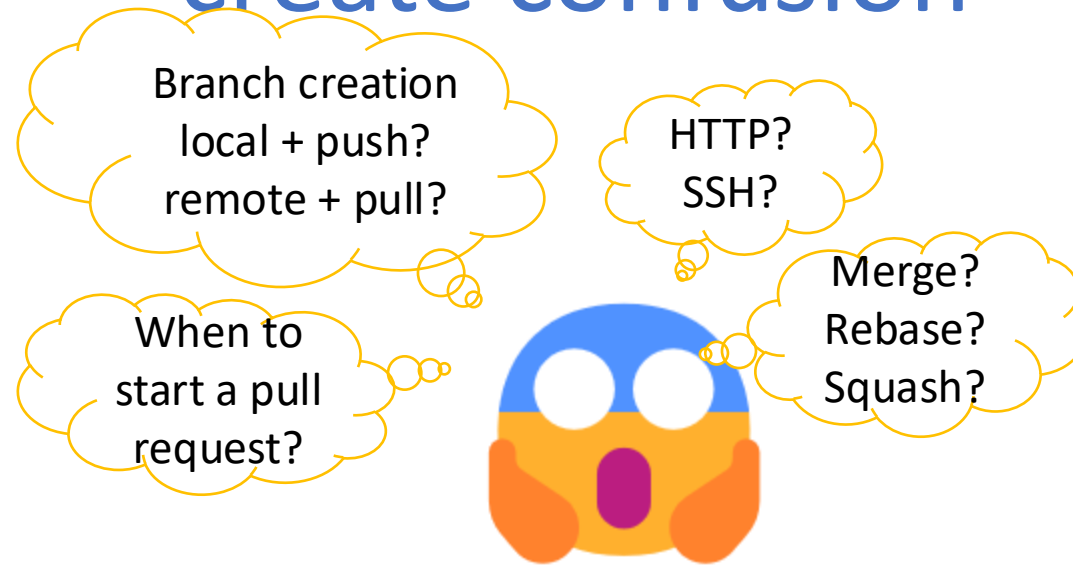


Videos



Annotated screenshots

Too many choices in git and GitHub create confusion



Choose a workflow for the community and document it.

Review process is very different from SVN and needs special attention. The suggestion feature is powerful when people know about it.

On pull requests:

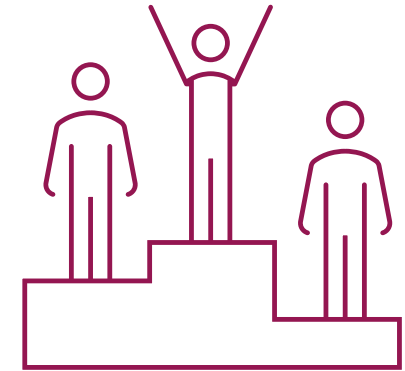
We ask pull requests to be open from the start. It helps visualize active development and promotes collaboration.



THE BENEFITS

Learning industry standard tools is best for workforce development

Attractivity of technical positions is important to draw in capable people



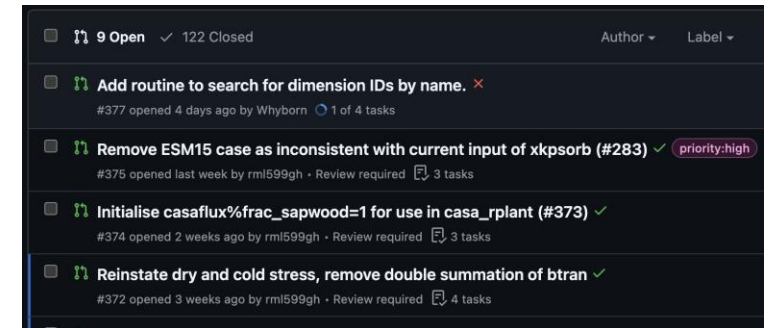
Students will have all sorts of careers; they need as many transferrable skills as possible



Git and GitHub are tailored for collaboration

On GitHub, all the information is “public” and searchable.

In particular, a list of open pull requests is easy to scroll through. This promotes spontaneous collaboration.



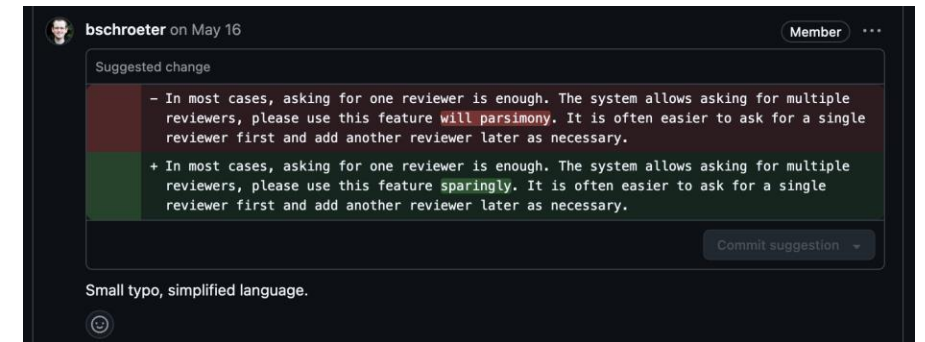
Open pull requests

All branches are by default open to all collaborators. Anyone can come and provide direct input at any time.

Comments attached to code and permalinks!

These simplifies documenting an issue and collaborating on implementation.

Suggestions in reviews speed up the process.



Code suggestion example



Permalink example

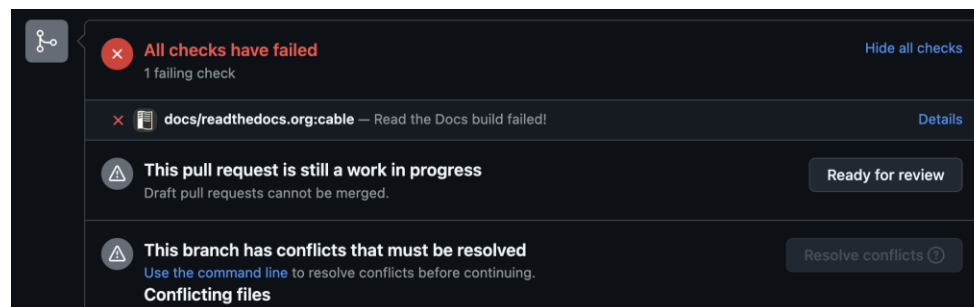
Streamlining workflows for reviews and testing

With review comments attached to the code modifications, it is simpler to know what the comment is about. Reviews also have **threaded discussions** in the reviews.

```
core/biogeochem/POPLUC.F90
...   ...   @@ -665,9 +687,6 @@ SUBROUTINE POPLUCStep(POPLUC,year)
665 687       if (POPLUC%ptog(g) .gt. 0.0_dp) &
666 688           CALL execute_luc_event('PRIMF','C3ANN',POPLUC%ptog(g),g,POPLUC)
667 689
668 -         ! if (POPLUC%stog(g) .gt.0.0_dp) &
669 -         !     CALL execute_luc_event('SECF','C3ANN',POPLUC%stog(g),g,POPLUC)
670 -
671 690       if (POPLUC%gtop(g) .gt.0.0_dp) &
```

har917 3 weeks ago Member ...
Not the topic of this particular PR - but should we remove %gtop from the code base entirely?

juergenknauer 3 weeks ago Member Author ...
Fine with me. I think the logic that nothing can be converted to p is reasonable.



GitHub blocks merging pull requests that have not met required conditions (reviews, tests, absence of conflicts).

GitHub also has **draft pull requests**: get the advantage of continuous integration testing from the start of your work!



Moving to git: short-term challenges for long-term benefits

We encountered the [common challenges to learning something new](#).
With the right support and community self-help, these challenges are [short-term](#).

[Increased, simpler collaboration and higher searchability](#) are the main benefits seen by the developers.

From the maintainer's side, integration with development tools, easier access to history and integrated testing [streamlines workflows](#).

