Peer-review process for IN4/3110

October 2, 2019

1 Introduction

Well done on finishing assignment 4! The next step of this assignment is a peer-review. The peer-review reflect how professional software development is actually performed today.

Consider the following example. Sarah owns a project repository and Bill would like to suggest an improvement or new feature to that project. The procedure for Bill's code contribution consists of five steps:

1. Bill creates a *copy* of Sarah's repository. This copy is called a 'fork'.

2. Bill implements his improvements or new feature and pushes these changes to his 'fork'.

3. Bill send a request to Sarah to include his changes. He does this by creating a 'pull request'.

4. Sarah reviews Bill's changes and can either reject or accept the pull request. If rejected, Bill can commit further improvement to his fork until Sarah is happy.

5. Once the pull request is accepted, Sarah merges the pull request. This means that Bill's code changes will be merged inot Sarah's code repository.

The figure below visualizes these steps. Note that this procedure even works if Bill does not have write access to Sarah's repository.
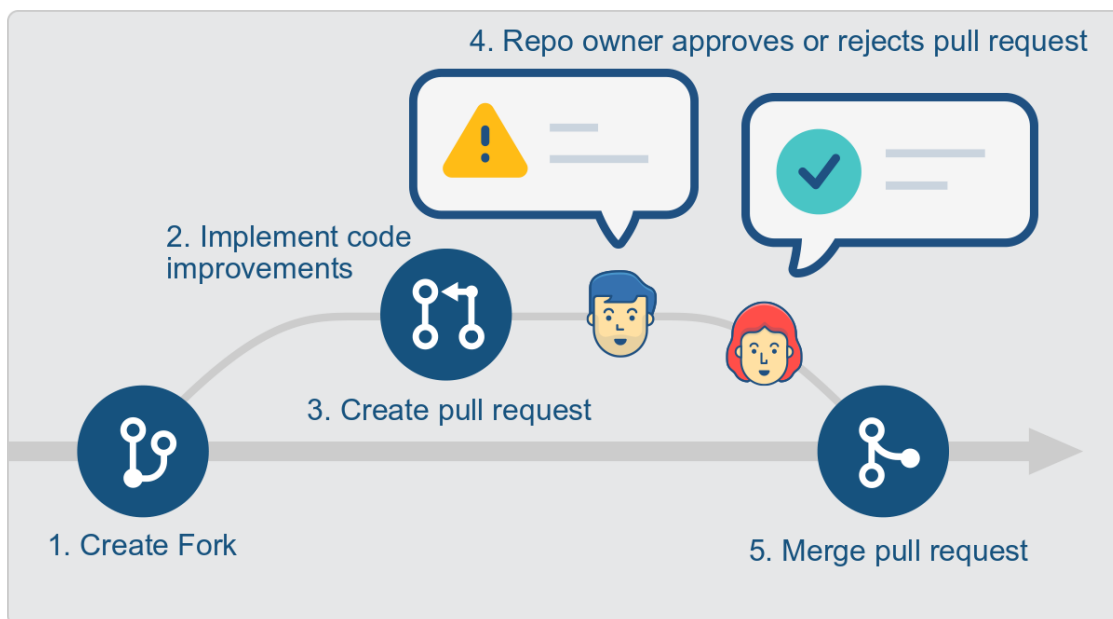

Figure 1: Steps of modern collaborative software development

1.1 Goal

In this peer-review we will perform steps 1-3 of the steps described in Figure **??**. That is, you will create a *copy* of the repo (a 'fork') and implement your improvements directly to the code. Once finished, you will request to include your improvements into the students original repo (a 'pull request'). This 'fork-then-pull-request' is a very common practice in professional code development. The spirit of the "review" is: *make constructive changes* and use *your knowledge and experience* to improve the students solution.

You will still be assigned to groups of 2-3, and it is up to you how you distribute the work for the peer-review (for instance each of you could be responsible for one pull-request each). You should help each other in the group to improve each other pull requests as much as possible, but you will only be graded for your own pull request.

The following list guides you through this process:

1. Visit the repository URL that you need to review, i.e. something like
   `https://github.uio.no/UiO-IN3110/IN3110-UiOUser`.

2. Click on the 'Fork' button on the top right to create your personal copy of the repository.
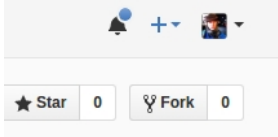
3. Once forked, you will be forwarded to the repository page of your personal copy.
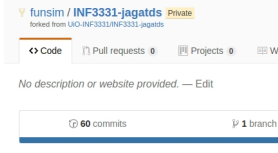
4. Clone this repository as usual to your computer.
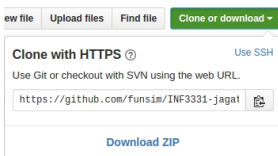
5. Review the code and implement improvements where possible. Commit and push these changes to your forked repository. Follow the guidelines in section **??** as a guideline. In case the student did not implement an assignment, you can skip the review of that assignment.

6. Once you are finished and committed the improvements, it is time to create a 'pull request'. This will notify the reviewed student that you would like to apply your changes to the student's code base. You create the peer-review by visiting your for repository (i.e. `https://github.uio.no/YourGithubUsername/IN3110-UiOUser`) and clicking on 'New pull request':
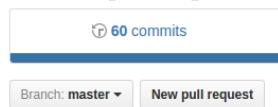
   In the description of the pull-request you should summarize and motivate the changes that you have made, as well as insert comments for specific parts of the code.

   More information about pull requests can be found on `https://help.github.com/articles/about-pull-requests`.

1.2   Guidelines

For each (coding) exercise, you should try to address and implement improvements to the following points:

- Add docstrings where missing and where appropriate.

- Is the code working as expected? For non-internal functions (in particular for scripts that are run from the command-line), does the program handle invalid inputs sensibly? Can you make the code more reliable?

- Is part of the code unreadable or difficult to understand? Simplify the code, add comments where required, and use classes/functions to avoid duplicate code.

- Do **not** commit suggestions on how to improve code quality to the code - instead actually implement these changes.

Remember: you should <u>implement improvements</u> in the pull-request and not just add comments into the code)!