Peer-Review Assignment 4

Introduction to Real Life Software Development

IN3110/IN4110 10 Points

University of Oslo - IN3110/IN4110

Fall 2021

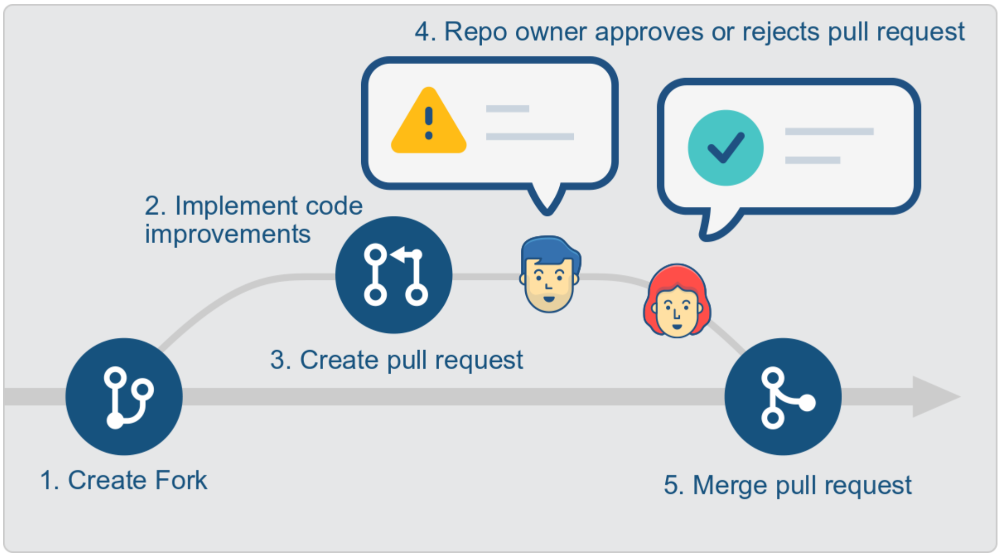
1 Introduction

Well done on finishing assignment 4! This assignment will be different than the ones before. Instead of writing your own code from scratch, we are going to peer-review the code already developed in assignment 4. This will reflect how professional software development is actually performed today. Specifically, in collaborative software development one does typically not write reviews in LaTeXto suggest code improvements.

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 sends 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 into 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. Also, this is exactly what you did in assignment 1 with your own repository.



4. Repo owner approves or rejects pull request
2. Implement code improvements
3. Create pull request
1. Create Fork
5. Merge pull request

Figure 1: Steps in collaborative software development

2 Goal

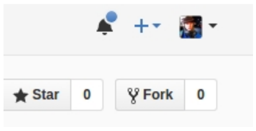
In this peer-review we will perform steps 1-3 of the steps described in Figure 1. That is, you will create a *copy* of the repo (a 'fork') and implement your improvements directly to the code. Once finished, you will request to include your improvements into the students original repo (a 'pull request').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

The idea of the "review" is as follows: *make constructive changes* and use *your knowledge and experience* to improve the solution. You will actually implement code improvements. You will be assigned randomly one repository and are responsible for one pull-request.

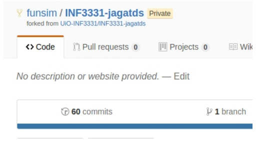
The following list guides you through this process:

1. Visit the repository URL that you need to review, i.e. something like `https://github.com/UiO-INF3331/INF3331-UiOUser`.

2. Click on the 'Fork' button on the top right to create your personal copy of the repository.



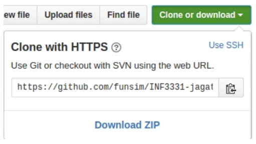
3. Once forked, you will be forwarded to the repository page of your personal copy.



No description or website provided. — Edit

60 commits
1 branch

4. Clone this repository as usual to your computer.



Clone with HTTPS

Use SSH

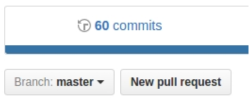
Use Git or checkout with SVN using the web URL.

`https://github.com/funsim/INF3331-jagat`

Download ZIP

5. Review the code and implement improvements where possible. Commit and push these changes to your forked repository. Follow the guidelines in section 3 for further instructions. In case the the repository you randomly got assigned is empty or does not contain the implemented assignment, you need to contact oscaro@simula.no or lisa@simula.no in order to receive a new repository to work on.

6. Once you are finished and have 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 the assigned repository (i.e. `https://github.com/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`.

3 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? Does the code need to handle further exceptions that are not yet captured?

- 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 - you have to actually implement these changes.
- Leave a summarizing message of your changes when submitting the pull-request!
-

Again: You should implement improvements in a pull-request and not just add suggestions into the code!

Note 1: You can only receive points for the peer-review if you submitted assignment 4!

Note 2: If you asked for an extension, you will be assigned a repository after your extension deadline. Have fun peer reviewing!

Q&A

- *Should we implement missing parts?* - Yes, if that improves the package (e.g. the user interface), you should definitely implement it.
- *If I get my repo later because I got an empty one, do i get extra time for the assignment?* - You can apply for a 3-day extension using the Google Form.
- *Are we expected to review the tasks given for IN4110, even if we have not done them ourselves?* - We recommend that you review all tasks that the student did (even if it is a IN4110 only task and you are a IN3110 student), but we will not be strict about it if you skip it. You do not need to review 4.0 (Profiling Warm-Up).
- *What do we do if the code we are reviewing is well documented, work as expected, handle invalid inputs, is easy to read, uses utilities functions to avoid duplicate code and (at least seem to) apply all good practices we've learned? I can't really find any improvements to implement ...* In case you received a perfect assignment to review, then the most important part of the assignment is to read it carefully and learn from it.

  Some ideas for improvements:

  - If there is potential to increase readability by choosing better function/variable names, then you should absolutely do that.
  - You can try and run the pep8 tool (https://www.python.org/dev/peps/pep-0008/). It automatically checks if standard Python syntax guidelines are followed.
  - If even then you do not see any improvements to be made, then just deliver an empty (or trivial) PR with a congratulation message to the author.
- *Should we implement structural changes as well, or simply try to improve each script in the repo we're handed?* - If this improves the package you were handed, you can also change the structure and argue for why you think it's a better structure.