Assignment 3

FIT2107

Reflection report

Done by:
Sutulova Tatiana 30806151

1. Introduction

Code review is a stage of software development in which the author of the code and other independent programmer(s) and quality assurance testers gather to review the code searching for defects in the program functionality and code quality. This activity is considered as relatively inexpensive but efficient and useful way to find and eliminate the bugs, which may result in huge issues in the future stages. The line by line code inspection allows the reviewers to check for flaws or potential flaws, consistency of the overall design, the quality of documentation and adherence to coding standards.

2. Results of the code review

There were two types of code reviews that were conducted on our code: the individual review, in which we, as the code developers, were reviewing our own code according to our checklists and the peer review, that allowed us to hear the feedback about our code from the peer programmers, who were not taking part in the development of this program.

The code review approach and the checklist that our team was using was divergent from our peers' ones, which helped us to cover the wider range of issues existing in our software. While reviewing we were mostly focusing on the quality of our code and comments to make sure the code was consistent, maintainable and efficient in terms of time complexity, whereas the main focus of our peers was to identify the issues by Blackbox testing the code and find what part of the code was causing the issue identified during testing.

The difference in code approaches assisted us in discovering the issues affecting the feature completion, side effects and test cases as well as violation of good programming practices including code smells and code duplication. Besides, improving the quality of the Calendar application, hearing from coders with completely different coding style helped us to learn new skills as well eliminate problems in our own coding habits.

3. Checklists improvements

As it was mentioned before the code review conducted by the other group was adding the analysis of the actual functionalities that were missing in our own review. Therefore, without their addition our checklist might be considered incomplete. In order to improve our checklists, the following points could be added:

- Are all the functionalities complete and give the expected output?
- Are there any redundant functionalities that may be removed?
- Do all functionalities follow the requirements?

Discussing about the possible improvements that our peer reviewers could make to their checklist, it must be stated that their entire emphasis was on functionality and, thus, a lot of significant problems related to the code quality were ignored. To improve their checklist, I would suggest them to research about the coding standards and best practices and make sure to include questions related to it into their checklist.

4. Peer review vs individual review

It is vital to mention that both peer and individual review had its own advantages and disadvantages. To begin with, individual code review was beneficial since the developer and the reviewer had a certain level of mutual understanding and social connection between each other, which increased the efficiency of the review activity. Thus, besides conducting a brief review that points out the most obvious issues, I, as a reviewer, was able to build a deeper

understanding of the code and identify the logical and structural errors. Moreover, the second advantage of individual review is lack of conflict. While conducting our own review on other group's code we faced the problem that developers of the code could take the issues that we found in their code personally, which resulted in misunderstandings and tension between reviewer and developer teams.

As for the peer review, the main advantage is that each participant of the code review may gain knowledge from their peers and at the same time impart their own skills and experience. Members of the team that was reviewing our code were more experienced programmers than us, which made us learn from them and helped us to eliminate problems in our programming behaviours. Besides that, the other benefit of the peer review is an independent view on the software. A reviewer has not taken part in the software development and therefore does not have any emotional attachment to it, which helps them to judge the code fairly, relying on the coding standards and requirements only.

5. Proposing fixes for found defects

Discussion whether the code reviewer should propose the possible solutions for the issues detected is a controversial topic, since it has its own pros and cons. Firstly, the reviewer is mostly focused on tracking the code quality more than checking the appropriateness of functionalities. Hence, it may be difficult for them to suggest a possible improvement since it is time consuming to understand the logic of complex structures. In this case, the problem must be properly explained making sure that developer has a complete understanding of it and since the author of the code has a proper awareness of their code it would be more efficient to find a solution themselves.

On the contrary, proposing fixes for the identified defects may be an essential part of explaining it, because sometimes it is easier to describe the problem by giving an example how to fix it. The proposed solution may help the coder to gain new knowledge as well, because the issue might be caused by the lack of skills and it would help to suggest an improvement more than just point where the problem is.

6. Conclusion

The code review activity was an important experience, since it allowed me to understand what problems there were in my coding style, that were leading to significant failures in the future stages of development and its functionality. Reviewers shared knowledge about programming practices and the code base that I was not aware about before due to lack of experience. In my opinion, it might be useful to conduct peer review on other artefacts, such as UML diagrams or documentation, since the independent expert is always able to critically and fairly examine my work based on the standard criteria.