Deliverable 3

August 2, 2019

Code Review of F5

1 Overview

The code review has been conducted of the Gamma Function program. Manual review has been done as well as automatic review has been done using CheckStyle Plugin.

2 Code Formatting

There are the following points that has been observed while going through the code

- Use of alignment (left margin) has not been followed properly.
- There are a lot of extra white spaces in the code.
- Proper naming conventions (Pascal, CamelCase etc.) have been followed.
- There are unnecessary comments in the code.
- The readability of the code is very complex because the names of the function should be descriptive about the nature of the functionality being implemented in the method.
- There are 1118 Checkstyle violation markers has been found in the code
- The code is exceeding the limit of average LOC per class. We can see that the code is more than of 400 lines.

3 Architecture

It does not follow any separation of concerns principle. An architecture should have been implemented because of the maintainability purpose.

4 Programming Code Style

- The code was seen to have a lot of hard coded values. Enums or property files should have been used for hard coded values
- It can be seen there are many if/else clauses for single line of statement. Ternary operators should have been used instead of that.
- There are some meaningful documentation in source code.

5 Non-Functional Requirements

- The same code has been repeated more than twice.
- Since the class file has more than 400 LOC henceforth it is tough to analyze and maintainability is affected.

6 Quality attributes

- The program implements unnecessary required functionality. Addition, Subtraction, Multiplication and Division implementations have been done that was not required which is adding the complexity of the code.
- The program is functionally correct.
- The program provides preventative error handling for user. It prompts message for complex operation.

Testing Review of F6

7 Testing Results

- Junit5 framework has been used for testcases.
- All the test cases have been passed.
- The bulit-in function 'assertEquals' has been used to check all the responses.

8 References

- Gee, T. (2019). Code review best practices — The ultimate software development tool. [online] Fogbugz.com. Available at: https://www.fogbugz.com/blog/code-review-best-practices/ [Accessed 2 Aug. 2019].
- Evoke Technologies. (2019). Code Review Checklist To Perform Effective Code Reviews. [online] Available at: https://www.evoketechnologies.com/blog/code-review-checklist-perform-effective-code-reviews/ [Accessed 2 Aug. 2019].
- GitHub. (2019). saadkhan2005/SOEN6011Project. [online] Available at: https://github.com/saadkhan2005/SOEN6011Project [Accessed 29 Jul. 2019].