Assignment 2 – Evaluating code quality through code reviews - Group assignment (3-4 members)

The goal of this assignment is to understand the subjectivity of code quality. The objective is to evaluate the code quality of an open source software using two code review tools. Quality is not always white and black; it is subjective which a tool cannot provide. Therefore, you will compare the outcome of the tools and provide your own subjective reflection through manual evaluation of the code quality. In this assignment you will create a review checklist based on the knowledge from Assignment 1, evaluate the quality of the code and evaluate the usefulness and accuracy of the tool.

The tasks in this assignment are as follows –

Select an open source software. The criteria for selecting the open source software should be as follows –

i. You **must** select the latest working software version.
ii. The code should contain 10-20 classes.
iii. It should be from a domain that is common and easily understood.

1. Select two code review tools. The criteria for selecting tool is as follows –
   i. It **must** be a free tool without any license cost.
   ii. It should be a widely used tool (check the tool reviews).
2. Create a checklist of the things you would like to consider in the code review. Use the input from the previous assignment and additional knowledge (online checklists) to create the checklist.
3. Evaluate the code manually and by using the tools.
4. Compare the outcome of the tools.
5. Compare manual evaluation and evaluation done by the tools.

The report should contain the following information in **separate sections,** the percentage in the bracket indicates the weightage of each section in the assignment grading/assessment –

- The link to the open source software and details (no. of classes, etc.) should be mentioned (**Mandatory**).
- A code review checklist (30%) – The checklist should consist of review themes that you would like to evaluate. For example, code formatting, architecture, coding standards and so on. Under each theme you should provide review questions for example, is the function or class too big. Provide references to the research articles from assignment 1 that mentioned the review aspect/theme/defect/question.
- Reporting, comparison and reflections of the tool outcomes (30%) – Report the values from both the tools, Are the tools reporting true or real values/measurement? Are values reported by the tools same or different values? Why is there a difference if any? Do the tools individually cover all the aspects in the checklist you created in Task 3?
- Final code review report (40%) – You should present the evaluation of the code by considering the outcomes of the tools and your manual evaluation. It is important to not just report values of the evaluation instead, you should elaborate why you think it is good or bad. For example, if the class is too long, does it have to be long? What implications does long class have on maintainability, security, etc.? Is it possible to divide the class into smaller classes? Are there many problems in a particular class? If so what could be the reason? etc. Consider that you are writing a code review for a developer who will improve the code based on your review. Therefore, the code review report should be easily comprehendible.