**Assignment 3: Evaluation and Comparative Analysis of Object Oriented Design in Open-Source Software**

***Objective***

This assignment aims to evaluate various aspects of object-oriented design in the selected open source software project using specific metrics and perform a comparative analysis of design patterns versus non-design pattern implementations.

***Task****:*

**1. Review Previous Work:**

• Review your previous assignments where you identified and extended design patterns in an open-source software project.

• Ensure you clearly understand the existing design patterns and their implementations. **2. Evaluate Object-Oriented Design:**

• Use tools like JDepend, SonarQube, Structure101, Lattix, CodeMR, NDepend, etc., to evaluate various aspects of the selected project.

• Calculate the following metrics for the project:

o Complexity Metrics:

▪ Cyclomatic Complexity

o Coupling Metrics:

▪ Coupling Between Objects (CBO)

▪ Afferent Couplings (Ca)

▪ Efferent Couplings (Ce)

o Cohesion Metrics:

▪ Lack of Cohesion of Methods (LCOM)

▪ Cohesion of Methods (CoM)

o Inheritance Metrics:

▪ Depth of Inheritance Tree (DIT)

▪ Number of Children (NOC)

o Encapsulation Metrics:

▪ Measure of Aggregation (MOA)

o Abstraction Metrics:

▪ Abstractness (A)

▪ Instability (I)

o Design Quality Metrics:

▪ Distance from the Main Sequence (D)

o Size Metrics:

▪ Lines of Code (LOC)

▪ Number of Methods (NOM)

o Responsibility Metrics:

▪ Weighted Methods per Class (WMC)

▪ Response For a Class (RFC)

o Maintainability Metrics:

▪ Maintainability Index (MI)

**3. Redesign Without Design Patterns:**

• Select two code regions where design patterns have been applied.

• Redesign these code regions without using the identified design patterns. • Ensure that the redesign maintains the original functionality and meets the project’s requirements.

**4. Perform Comparative Analysis:**

• Evaluate the redesigned code regions using the same metrics listed above. • Compare the results of the metrics for the original design (with design patterns) and the redesigned implementation (without design patterns).

**5. Document Your Work:**

o Prepare a detailed report that includes the following sections:

1. **Introduction**: Briefly describe the original project and the design patterns identified and extended in the previous assignments.

2. **Design Evaluation**: Present the results of the object-oriented design evaluation using the specified metrics. Include charts or tables to illustrate the metrics.

3. **Redesign Description**: Describe the redesign of the two selected code regions without using design patterns.

4. **Comparative Analysis**: Provide a comparative analysis of the original and redesigned code regions using the metrics. Discuss the differences in complexity, coupling, cohesion, inheritance, encapsulation, abstraction, design quality, size, responsibility, and maintainability.

5. **Conclusion**: Summarize your findings and insights gained from the evaluation and comparative analysis.

**6. Submission:**

• Submit your report as a PDF document.

• If necessary, include any supplementary materials, such as updated class diagrams, code files, or additional documentation.

***Evaluation Criteria***

• **Analysis Design Evaluation:** Thoroughness and accuracy of the evaluation of object oriented design using the specified metrics.

• **Redesign Quality:** Quality and correctness of the redesigned code regions without using design patterns.

• **Comparative Analysis:** Depth and clarity of the comparative analysis, including the use of metrics to support conclusions.

• **Report Quality:** Overall quality of the report, including structure, coherence, and presentation.

***Deadline***

• Submit your assignment by 05-07-2024.

By completing this assignment, you will gain a comprehensive understanding of various software quality metrics, enhance your skills in software design, and appreciate the impact of design patterns on software quality and maintainability.