Lab 2 – Software Quality

Software Architecture and Design

Student Name: Kaushalkumar sharma
Lab: 2 – Software Quality

1. Introduction

This lab is aimed at examining the qualities of software and more specifically the concept of modularity and its impact on the concept of coupling, cohesion, and modifiability. Modularity enables the complex systems to be divisible into autonomous and interchangeable component parts, enhancing maintainability, testability, as well as long-term evolution.

2. Selected Systems and Size Validation

The list of open-source software systems provided by the instructors was used to choose four systems. The sizes and complexity of the systems were compared to make an even comparison, through the validation of similarities between the systems through the use of Source Lines of Code (SLOC), number of classes, modules, and number of commit. The systems were also found to be comparable and therefore analysis was possible.

3. Metrics and Tools Use

Extracting metrics was done using the instructor-provided script class metrics v6.py.

Cohesion Metrics: The lack of cohesiveness in methods. - TCC (Tight Class Cohesion)

Coupling Metrics: CBO (Coupling Between Objects) is used to define the relationship between objects.<|human|>CBO (Coupling Between Objects) is a concept that is used to describe the connection between objects.

- Fan-In

- Fan-Out

Modifiability Metrics:

- Class Change Frequency –

Change in Lines per commit (NLC)

4. Data Collection Process

The command that was run for every project was: py class_metrics_git_v6 repository-path output-filename. It makes use of Python history analysis (ast parsing and Git history analyzing) and Python dependency-based analysis (pyreverse) to do all the static analysis.

5. Data Analysis and Visualization

The summary statistics, histograms and cumulative distribution functions (CDFs) were produced with the help of Excel. Cohesion Analysis revealed that low LCOM values are associated with the high TCC values. The Coupling Analysis showed that high CBO could be accompanied with a high fan-out. Modifiability Analysis indicated that the more the classes are coupled the more they are changed.

6. Cross-System Comparison

Lowerly coupled and highly cohesive systems were always found to be more modifiable and required less maintenance effort.

7.Summary of Findings

The results verify that modularity is a strong contributor to the quality of software. Tightly coupled and loosely coupled classes are less difficult to maintain and evolve.

8. Conclusion

The lab supports the fact that the significance of modular design is great to ensure software quality and sustainability..