

ABSTRACT

Background & Problem

Software maintainability is expensive and time-intensive. It involves tracking the changes to over time. The process involves software evolution analysis. If software is not analyzed, software maintenance become more time-consuming & expensive.

Approach

In this research, we collect the five code repositories over the course of six years from 2016-2021. The repositories are thoroughly analyzed using IntelliJ tool integrated with CodeMR tool.

Results

The results revealed that there is a correlation between high code complexity and difficulty for software maintainability.

CONTACT

Giselle Gonzalez

Pasadena City College

gelly.m.gonzalez@gmail.com

INTRODUCTION

- Software maintainability is a time-consuming and expensive task in the software development process.
- There is no standard software design to ensure a better software code maintenance.
- However, software evolution may find out why software maintenance becomes more complex.

RESEARCH METHOD

- Six years versions of five different software repositories are selected
- Extract the code quality metrics for each version
- Analyze the evolution of software code quality to draw the patterns over time.
- Find out the evolution of the code quality metrics.
- Find out the complex classes against code quality metrics to get an idea of software maintenance.

RESULTS

The first result shows that High complex classes are lower compared to low complex classes visualized in Fig 1. This is a good sign for future software maintainability.

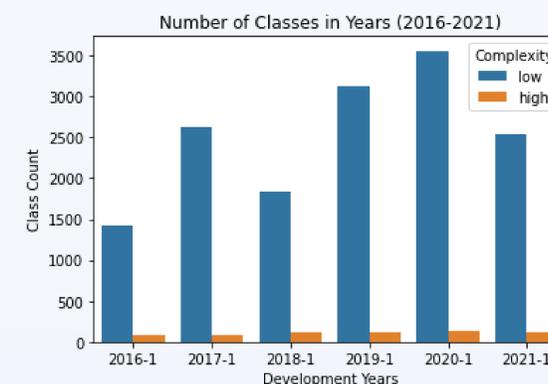


Figure 1: Class number evolution by class complexity

Evolution of ATFD (Access to Foreign Data) values in high complex classes are linked with a higher probability of becoming a GOD class code smell. Except for 2016-17, the ATFD value increased each year in the high complex classes shown in Fig 2.

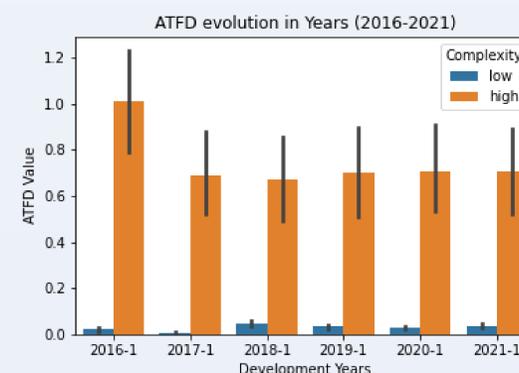


Figure 2: ATFD evolution by class complexity

The relationship between LOC (Lines of Code) and WMC (Weighted Methods Per Class) is used to determine software complexity. The higher levels of WMC and LOC metrics appeared in later years as the software developed.

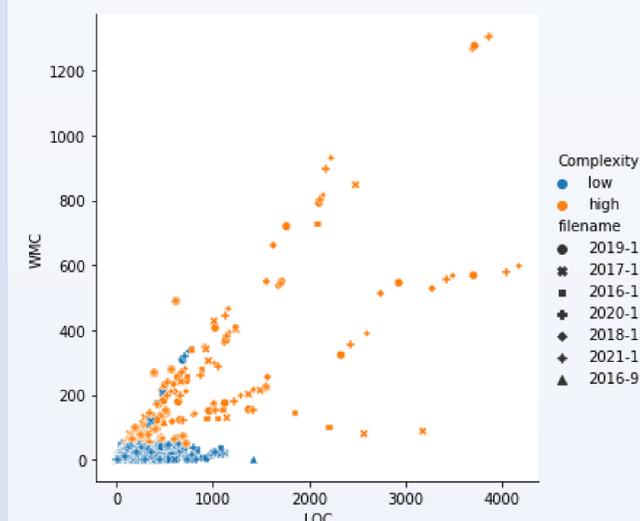


Figure 3: Distribution of LOC & WMC by complexity

CONCLUSIONS

- Code quality metrics are very helpful to find out highly complex classes for maintenance.
- Software maintenance becomes more easier with the software code quality evolution through code quality metrics.