Grant Proposal

Jincheng He
Department of Computer Science
University of Southern California
jinchenh@usc.edu

I. Introduction

Developing software with the source code open to the public is very common. However, similar to its counterpart, open-source software also has quality problems, causing functional failures, such as and unsatisfying user experience, such as long responding time. To improve the quality of open source software, we investigate how the quality is impacted by commits of different purposes. By identifying these impacts, we will establish a new set of guidelines for commiting changes, thus improving the quality.

II. Previous work

Previous researchers have revealed when, where, how and what the developers contribute to projects and how these aspects impact software quality. However, there has been little work on how different categories of commits impact software quality.

III. Our Plan

A. Stage One

Previous researchers have not studied the correlation between the change type and the code or their impact on quality. In this stage, we start with refining existing categorization of commit changes in open source software repositories. We evaluate the quality of those changes by obtaining quality metrics from static analysis tools. To assess the correlation between the quality and the categories, we plan to train a machine learning model, in addition to applying standard mathematical correlation analyses.

B. Stage Two

In this stage, although we have categorized the commit changes, further work distinguishing between different categories is required. This is because it has been shown that high-level categories have overlaps with each other. In this stage, we will remove the ambiguity of the categories by analyzing the code changes within the commits rather than the commit messages and manual categorizing. Once this is done, we will investigate the correlation between the categories and changes in code to reveal whether they correlate with each other and how those changes impact software quality.

C. Stage Three

In the final stage, with the refined categories, we will construct guidelines for developers on how they can better contribute to open source software when they make different types of changes. In addition, we will create an index to indicate how different code patterns impact the quality. We will conclude this research by completing and validating these two aspects.

IV. Feasibility

This project is feasible according to our investigation in the following three aspects:

- Data: Open-source software and version control systems provide sufficient meta-data for analysis.
- Tool: Existing tools, such PMD, SonarQube, FindBugs and CAST provide various quality metrics.
- Techniques: The machine learning and natural language methods required by this research already exist.

With all above provided, we believe this plan will succeed in 3 to 5 years. The midterm milestone is obtaining a reasonable high prediction accuracy from the machine learning model. The final milestone is the completion of the new systematic coding standard and development guidelines.

V. Benefits

We will be able to provide guidelines on how open-source software developers, when contributing to projects, can attain better quality. In addition, the results in the second stage will allow us to provide more reliable coding standards and will improve the overall code quality. Improved quality will either help to reduce the cost or improve the software service quality.

VI. Intellectual Advancement

The first goal of this research is to practically improve the quality of open-software by providing guidelines for developers about how to better contribute to projects and to improve code quality. We believe this will be an intellectual improvement in software engineering since it will change the way people think, code and develop software.