

Grant Proposal

Jincheng He

Department of Computer Science

University of Southern California

jinchenh@usc.edu

I. INTRODUCTION

This research investigates how the different purposes of commits impact the quality of software open source software (OSS).

Furthermore, this research investigates in a more atomic level of how different commits, with respect to their purposes, differ in code pattern and how can we improve the way people code based on the results, consequently contributing to a better code quality.

II. PREVIOUS WORK

In this direction, research has been conducted to investigate when, where, how and what the developers contribute to the projects and how they may impact software quality. However, there is little work focusing on purpose-related categorizations and how different categories may impact software quality.

III. OUR PLAN

A. Stage One

Research has been done to investigate the impact of some behaviors of developers when they contribute to software, especially open source software. However, in the previous research in this area, they haven't reveal the correlation between the change type and the code as well as their impact on the software quality. In this stage, we start with categorization of commit changes in open source software repositories and show their impact on the software quality and validate the categories by prediction models.

B. Stage Two

In this stage, as we have established a categorization, we will be working on how we can improve the quality of the categorization, which means having a more convincing categorization that represents commit changes more accurately. Once this is done. We will investigate the relation between the categorizations and atomic changes in code to reveal whether they have close connection and how the atomic changes impact software quality.

C. Stage Three

Once the categorization and its relation with atomic changes in code are revealed, we will dive into the next stage which is guidelines for developers about how to better contribute to an open-source software (and write code) and a prediction model for predicting categorization.

IV. FEASIBILITY

- 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 in this research already exist.

With all above provided, we believe this plan will succeed in 3 5 years.

V. BENEFITS

Once the plan work out, we will be able to provide guidelines on how open-source software developers contribute to projects to attain better quality. In addition, results in the second stage will allow us to provide more reliable coding standards which will improve the overall code quality.

VI. TIMELINE AND MILESTONES

This research will take 3 to 5 years for these three stages, although this direction may take more than ten years since changing standards is a gradual process. The success of this direction depends on how the applied techniques evolve in the coming years. The midterm milestone could be a reasonable high prediction accuracy while the final exam milestone could be the completion of the new systematic coding standards.

VII. INTELLECTUAL ADVANCEMENT

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