

Analyses of software project characteristics on pull request acceptance in distributed software development

Haoyu Wang

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfillment of the requirements of the Degree of Master of Science at the University of Glasgow

10th December 10,2021

**Abstract**

With the continued growth of the web and the advent of distributed version control systems, distributed software development has become a mainstream development approach, especially as social coding tools such as GitHub[1]have changed the way software is developed collaboratively and publicly on the World Wide Web [2]. Instead of pushing changes to a central repository, development developers are pulling them from other repositories and merging them locally [3]. This work builds on the dataset of Xunhui Zhang, Ayushi Rastogi, Yue Yu et al.'s study [4] to investigate the impact of item features in pull requests on whether pull requests can be successfully merged. There are six main project features, namely Programming languages, Popularity of project, Age of project, Workload of a project, Activeness of project, and Openness of a project. Using the project characteristics provided in the dataset [4], data cleaning as well as data analysis and data visualization were carried out to find out the relationship between project characteristics and the success rate of pull requests.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic form.

Name: Haoyu Wang Signature: Haoyu Wang

Acknowledgements

First, I would like to express the deepest appreciation to my supervisor, Handan Gul Calikli, for her all feedback, encouragement, and guidance throughout this project and for this amazing opportunity of studying on Data Analyses. Without her help, I cannot achieve what I have done.

Last but not least, I would like to thank all the participants, who are also my classmates and friends, for taking the time to give me help with the data analysis.

Additionally, I want to thank my parents for their unconditional supports and encouragement.

Contents

Chapter 1 Introduction 1

Chapter 2 Background 2

2.1 Pull Request 2

2.2 Relate work 2

Chapter 3 Tools and Methodology 3

3.1 Dataset 3

3.2 Tools 3

3.3 Measures or techniques 3

3.3.1 A subsection <This is style Heading 3> 3

3.3.2 A subsection <This is style Heading 3> 3

3.3.3 A subsection <This is style Heading 3> 3

Chapter 4 Results and Discussion 5

4.1 Programming languages 5

4.2 Popularity of project 5

4.3 Age of project 5

4.4 Workload of project 5

4.5 Activeness of project and Openness of project 6

Chapter 5 Conclusions 7

Reference 7

Appendix A <Name of appendix> 1

Appendix B <Another appendix> 2

# Introduction

<This is style Normal. We recommend you make use of styles to simplify creating a well-formatted document. We have used “space before” and “space after” in defining these styles, in order to space the headings and paragraphs appropriately. You should never need to enter a blank line.>

# Background

## Pull Request

## Relate work

<Each new chapter should appear on a new page.>

Background concepts (if required) and overview of relevant previous work (critically evaluate strengths and weaknesses).

# Tools and Methodology

## Dataset

## Tools

## Measures or techniques

### A subsection <This is style Heading 3>

### A subsection <This is style Heading 3>

### A subsection <This is style Heading 3>

The content of these chapters depends on the project and should be agreed with your supervisor (e.g., description of the solution, evaluation results, etc).

<Figure below is in style “figure” which continues to style “figure caption” when you press Enter and then back to “Normal” when you press Enter again.>

Figure 1: Some important shapes.

<If you wanted to show any code fragments, you could use the following style called code, which could then be followed by figure caption.>

*# This is a little bit of Python*

**for** i in range( 10 ):

**for** j in range( 10 ):

**print** i\*j,

**print**

Figure 2: A crucial algorithm for the project.

# Results and Discussion

## Programming languages

## Popularity of project

## Age of project

## Workload of project

Next is the impact of workload on Pull Request acceptance in the context of project characteristics. I have based this mainly on The Influence of Non-technical Factors on Code Review[5] and Determinants of pull-based development in the context of continuous integration[6], two papers that I have based my research on the approaches described in the papers.

形状, 矩形

描述已自动生成First, we tested the normality of our data by applying the Kolmogorov-Smirnov test[7]. It can be found that the *p*-value of workload the data is much less than 0.05, which indicates that the data is not normally distributed. We therefore used a non-parametric statistical test: the chi-square test[8] to test the correlation between workload and pull request acceptance.

1. Workload count

Before conducting the chi-square test, I first examined the distribution of the data. As shown in Figure 1, I found that most of the data were concentrated in the range of 0-500, so I further refined the data in the range of 0-500 and found that most of the data were concentrated in the range of 0-50, so I divided the data into 14 stages and counted the number of pull requests that were successfully merged and failed to be merged in each stage, as shown in Table 1.

1. Workload group



From the table we can see that the majority of pull requests are opened for uploads with a number of pull requests mainly between 0 and 200. The other parts do not account for much. We run a chi-square test on the data for the 14 quantity intervals based on the above table, where I assume the original hypothesis that workload and pull request acceptance are uncorrelated. After conducting the chi-square test, we can find that the p-value is much less than 0.05, which indicates that workload is significantly correlated with pull request acceptance. And from the table we can see the percentage of successful mergers versus the percentage of failures. As the number of pull requests opened during the pull request upload rises, the chance of the pull request acceptance is decreasing。

To further demonstrate our results, I used a multilevel mixed-effects logistic regression model to calculate coefficients and significance levels based on the approach described in the article Determinants of pull-based development in the context of continuous integration[6]. This is because the results for each pull request are dichotomous (i.e., merged\_or\_not).

图形用户界面, 表格

描述已自动生成

After simulations using the multi-level mixed effects logistic regression model, as shown in Figure 2，the p-value is less than 0.05. This further confirms our result that the success of workload and pull request merging is significantly correlated. I also found that the coefficient is less than 0, which indicates that this feature is negatively correlated with pull request acceptance. This suggests that the previous results are convincing.

## Activeness of project and Openness of project

Main conclusions of your project. Here you should also include suggestions for future work.

# Conclusions

# Reference

1. GitHub: Social Coding. [Online] Available: http://github.com/
2. Khadke, N., Teh, M. H., & Shen, M. (2012). Predicting acceptance of github pull requests. Stanford–CS 229, Tech. Rep.
3. Gousios, G., Pinzger, M., & Deursen, A. V. (2014, May). An exploratory study of the pull-based software development model. In Proceedings of the 36th International Conference on Software Engineering (pp. 345-355).
4. Zhang, X., Rastogi, A., & Yu, Y. (2020, June). On the Shoulders of Giants: A New Dataset for Pull-based Development Research. In Proceedings of the 17th International Conference on Mining Software Repositories (pp. 543-547).
5. Baysal O, Kononenko O, Holmes R, et al. The influence of non-technical factors on code review[C]//2013 20th working conference on reverse engineering (WCRE). IEEE, 2013: 122-131.
6. Yu Y, Yin G, Wang T, et al. Determinants of pull-based development in the context of continuous integration[J]. Science China Information Sciences, 2016, 59(8): 1-14.
7. Massey Jr F J. The Kolmogorov-Smirnov test for goodness of fit[J]. Journal of the American statistical Association, 1951, 46(253): 68-78
8. Chi-squared\_test :https://en.wikipedia.org/wiki/Chi-squared\_test

###### <Name of appendix>

<Use Heading 6 for the Appendix heading>

###### <Another appendix>