

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 & Related Work 2

2.1 Pull Request 2

2.2 Dataset & Project characteristics 2

2.2.1 Programming languages 2

2.2.2 Popularity of project 3

2.2.3 Age of project 3

2.2.4 Workload of project 3

2.2.5 Activeness of project 3

2.2.6 Openness of a project 4

2.3 Relate work 4

Chapter 3 Tools and Methodology 5

3.1 Dataset processing 5

3.2 Tools 5

3.3 Measures or techniques 5

3.3.1 A subsection <This is style Heading 3> 5

3.3.2 A subsection <This is style Heading 3> 5

3.3.3 A subsection <This is style Heading 3> 5

Chapter 4 Results and Discussion 7

4.1 Programming languages 7

4.2 Popularity of project 7

4.3 Age of project 7

4.4 Workload of project 7

4.5 Activeness of project and Openness of project 8

Chapter 5 Conclusions 9

Reference 9

Appendix A <Name of appendix> 1

Appendix B <Another appendix> 2

# Introduction

Github is a hosting platform for open source and private software projects. It provides a platform for software developers around the world to communicate and collaborate with each other. Any software developer can develop and open source their own projects on Github. As more and more developers recognise and enjoy distributed development, more and more projects, both open and closed source, are being migrated to code hosting sites like Github[3]. Github has implemented a distributed development model, called pull requests, for developers to be able to do pull-base development. The special feature of this distributed development model is that it allows any user to clone the public repository uploaded by the development team. This means that the user does not need to be part of the development team to make changes to the repository. Once modified, they can upload their modified code for review via a pull request. If the review is approved, the changes can be merged into the development team's repository and you can become a contributor to the project.

However, the existence of censorship means that a number of factors, such as contributor type, project language, development time and many others, can affect the pull request. The study by Zhang Xuhui et al.[4] builds on the study by Gousios et al.[3] by creating a new upgraded dataset called new\_pullreq. This dataset contains a total of 11230 items, 33347937 pull requests and 96 features [4]. The 96 features are divided into three broad categories: contributor-related, project-related, and pull request-related. There are features with intersections within the three categories.

The main objective of this paper is to replicate the correlation study between item characteristics and pull request merge success in the nine papers mentioned in the article by Zhang Xuhui et al. on item characteristics, and to build on their work to find one or more methods of data analysis to correlate parts of the item characteristics in the dataset. This led to the identification of features in the item characteristics that have a greater impact on pull request merging.

The structure of the paper is as follows: Chapter 2 provides a background to the paper, a basic explanation of pull requests, and a summary of the methods and conclusions of the related papers on the item features to be studied. Chapter 3 explains the characteristics of the items in the dataset and the partitioning and cleaning of the dataset. The tools and methods used in the analysis of the data are also explained to make it easier for others to copy the code when viewing the paper. Chapter 4 presents the results of the data analysis of the dataset and discusses them. Chapter 5 concludes with a summary of the whole project and a reflection on the results of future work.

# Background & Related Work

## Pull Request

As described in the introduction, pull request is a new distributed development model that works in the following way: first the core development team uploads their project to the main GitHub repository. When potential contributors want to make changes to the project, they don't do so directly in the main repository; instead, they fork a new repository that is independent of the main repository. Contributors can then make changes to the project on this new fork. When the change is made, it does not appear directly on the main repository, but the contributor generates a pull request and uploads it to GitHub, where it is reviewed by the core development team. If the change is functional and satisfactory, the pull request will be merged; if not, it will be rejected and the contributor can be asked to make more changes.

## Dataset & Project characteristics

The dataset used in this study was created using the upgraded dataset created by Xunhui Zhang et al[4]. This dataset contains 96 indicators. These 96 metrics are divided into three main categories in Xunhui Zhang et al.'s study. I focus on the project characterist, which contains a total of 7 metrics from the dataset, divided into 6 project characterist attributes.

### Programming languages

Six programming languages are included in the dataset: JavaScript, Java, Python, Ruby, Go and Scala, with different projects using different languages and accepting different rates of pull requests. The statistics for the 11,230 projects in the dataset are shown in Figure 1, which shows that JavaScript, Java and python are used more frequently in all projects. The indicator in the dataset is *language*[4]. 图表, 条形图

描述已自动生成

1. Programming language count

### Popularity of project

The popularity of a project is mainly determined by the number of forks that already exist in the project at the time of the pull request, the higher the number of forks, the more popular the project is. This is represented by *fork\_count*[4] in the dataset metrics.

### Age of project

The age of a project is the interval between when the project was created and when the pull request was created (month), and is represented in the dataset metrics using *project\_age*[4].

### Workload of project

The workload of a project is the number of pull requests in the queue waiting to be reviewed when the pull request is submitted, expressed as *open\_pr\_num*[4] in the dataset metrics.

### Activeness of project

The activeness of a projec is the interval between two pull requests when the last two pull requests for that item were uploaded(seconds). The shorter the time interval the more active the project is. It is represented by *pushed\_deltal*[4] in the dataset metrics.

### Openness of a project

The openness of the project is the number of issues that exist at the time of the pull request submission. The higher the number of issues, the higher the openness. The *open\_issue\_num*[4] is used in the dataset metrics to indicate this.

## Relate work

In recent years, there has been a growing body of research on the impact of project characteristics on the success rate of pull request mergers. In this section, we provide a review of research involving six characteristics of items to determine how we should replicate their research and propose a new data analysis method to investigate the relationship between item characteristics and pull request merge success rates.

• In a study by Soares D M et al.[5] on the impact of programming languages on pull request pairs, they calculated the programming language and merge lift by using association rules, and by comparing the lift of different programming languages, thus deriving the impact of different programming languages on the success rate of pull request merges. In contrast, in the study by Rahman M M et al.[6] the number of successful and failed merges of pull requests in different languages was counted by means of a double bar chart and compared to derive the guan of different programming languages in relation to the success rate of merges of pull requests.

•Similarly, in the study by Rahman M M et al[6], the effect of the number of project forks on pull request merging was studied, using the same methodology as when studying programming languages, by also looking at the effect of the number of forks on merging through a graphical statistical approach. In contrast, in the study by Khadke, N et al.[2] their main work was to predict the merge outcome of pull request requests by using a regression model, using several regression models for prediction, among the features they used was the number of forks. Although their aim is not the same as ours, it is also possible to obtain the correlation coefficient between the number of forks and the merge success rate when they use regression models to predict, and thus know whether there is a positive or negative correlation.

•In the studies by Tsay J et al.[7] and Yu Y et al.[8] they both used a multi-level mixed effects logistic regression model to predict the likelihood of accepting a pull request. The effect of item age on the success rate of pull request merging was derived by predicting the correlation coefficient and correlation in the model for the age of the item.

•Similarly, in the study of the effect of project workload on the success rate of pull request mergers, Yu Y et al.[8] used a multi-level mixed effects logistic regression model for correlation analysis, as did project age. In contrast to the above, in Baysal O et al.[9] they first used Kolmogorov-Smirnov tests to examine the distribution of the data and then used Kruskal-Wallis ANOVA to examine the correlation between project workload and pull request.

•For both project activity and project openness, which were used as essential characteristics in the study by Khadke, N. et al. logistic regression models were used to predict the merge success rate of pull requests. The correlation coefficients of these two characteristics on the merge success rate were obtained in the prediction process to derive the correlation with the merge success rate of pull request.

Taken together, these studies demonstrate that pull request acceptance and item characteristics are correlated. However, the above studies do not summarise all the correlations between the characteristics, so the next step in this study is to bring them all together and draw specific conclusions using data analysis.

# Tools and Methodology

## Dataset processing

## 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. Soares D M, de Lima Júnior M L, Murta L, et al. Acceptance factors of pull requests in open-source projects[C]//Proceedings of the 30th Annual ACM Symposium on Applied Computing. 2015: 1541-154
6. Rahman M M, Roy C K. An insight into the pull requests of github[C]//Proceedings of the 11th Working Conference on Mining Software Repositories. 2014: 364-367.
7. Tsay J, Dabbish L, Herbsleb J. Influence of social and technical factors for evaluating contribution in GitHub[C]//Proceedings of the 36th international conference on Software engineering. 2014: 356-366.
8. 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.
9. 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.
10. Massey Jr F J. The Kolmogorov-Smirnov test for goodness of fit[J]. Journal of the American statistical Association, 1951, 46(253): 68-78
11. Chi-squared\_test :https://en.wikipedia.org/wiki/Chi-squared\_test

###### <Name of appendix>

<Use Heading 6 for the Appendix heading>

###### <Another appendix>