Variability in Factors Influencing Pull Request Merge Decisions: A Microscopic Exploration

by

Nasif Ahmed

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Context: The pull-based development model is a widely adopted practice in distributed version control systems, particularly in open-source projects. In this model, contributors submit pull requests proposing changes to the codebase, which are then reviewed and potentially merged by project maintainers. Previous studies have extensively investigated the influence of different factors in merge outcome, aiming to generalize their impact across multiple projects. **Objective:** This thesis takes a unique approach by examining these factors at the project level, aiming to understand how the influence of each factor varies across projects. **Methodology:** To achieve this, we conducted a large-scale quantitative analysis on 841,399 pull requests from 1,100 GitHub projects. We constructed fixed-effect logistic regression models for each project and explored the correlations between different factors and merge outcomes. **Results:** Our analysis indicates that the influence of factors varies across projects, both in terms of their order and direction. For example, while contributor experience is highly valued in many projects, it was found to be statistically insignificant in others. Likewise, the likelihood of a successful merge increases with the number of commits in some projects, whereas in others, it has the opposite effect. These findings have implications for both researchers and practitioners.

Acknowledgements

I would like to express my deepest gratitude to:

Professor Mei Nagappan, for his invaluable guidance and understanding throughout my academic journey. He has consistently gone above and beyond his role as a supervisor.

Professor Weiyi Shang and Professor Shane McIntosh, for their constructive feedback and invaluable suggestions that have significantly contributed to shaping this work.

My parents, and my sister, for always keeping me in their prayers. Without their support and encouragement, I wouldn't have reached where I am today.

Most importantly, my beloved wife, Israt Jahan Ritun. Her unconditional love, boundless patience, and unwavering encouragement sustained me through the challenges of this academic journey. She has been an instrumental inspiration, and this degree is as much hers as it is mine.

Dedication

To my lovely daughter, Nusaybah.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The pull-based development model is a widely adopted paradigm for collaborative software development. In this model [11], contributors propose changes to the main repository by creating pull requests for integrators to review. The integrators then assess the proposed changes and, based on several factors, decide whether to accept or reject them. Understanding the factors that influence the merge decision of pull requests is crucial because it enables contributors to align their proposed changes with the expectations and criteria of the integrators. This alignment increases the likelihood of a successful merge, minimizes both the integrator and contributors' time and effort, ensures adherence to project standards, and promotes a more efficient development process.

Several studies have aimed to identify the factors influencing pull request merge decisions [28] [36] [37] and explore their relative importance [11] [41]. For example, in one study, Dey et al. [8] analyzed 483,988 pull requests from 4,218 npm projects to model the probability of a pull request getting accepted within a month of its creation. This study used a random forest model based on 50 influencing factors, categorized into developer, pull request, and project characteristics. Notably, out of these 50 factors, only 14 were sufficient to achieve high accuracy, indicating their significant influence on pull request acceptance decisions. In another study, Zhang et al. [41] analyzed 95 features from 3,347,937 pull requests in 11,230 projects to understand their relative relevance in merge decisions. Their results indicate that only a small subset of the factors (5 to 10) tends to influence pull request decisions, and their relative relevance also varies depending on context.

While previous studies primarily focused on the general understanding of the influences of different factors across projects [8] [11] [28] [36] [37] [41], it is also crucial to recognize the project-specific influences these factors have on merge decisions. The general influence of a factor on thousands of projects may not accurately reflect its impact on each specific project. Additionally, while a factor may have a positive influence on merge decisions in one project, it may have a negative influence on another project. For example, the factor "team size" may generally have a positive influence on merge decisions across projects. However, in a specific project where the team size is exceptionally large, it may have a negative influence on merge decisions due to coordination and communication challenges among team members. Conversely, in another project with smaller team size, the same factor may facilitate smoother communication and coordination among team members during the merging process, leading to fewer rejections.

As a result, from a developer's perspective, understanding the overall influential factors derived from thousands of projects may not hold for the specific project to which they are contributing. For them, it is more important to identify the most influential factors specific to their project, enabling them to tailor their pull requests based on those factors. By identifying the key contributors to the success of their project, developers can focus on meeting those specific requirements and guidelines. This targeted approach ensures that the pull requests align with the project's objectives, making it more likely that they will be accepted. Furthermore, understanding the project-specific influential factors allows developers to allocate their time and efforts efficiently, resulting in a more streamlined development process.

Given this context, this study focuses on understanding the project-specific factors that influence pull request merge decisions. More specifically, it addresses the following three intriguing research questions:

- *How does the order of influences of factors differ across projects and within projects?*
- *Do the same factors have different directions of influence in different projects?*
- *Which projects are similar in terms of the factors influencing their merge decisions?*

1.2 Contributions

In pursuit of answers to these questions, we analyzed 841,399 pull requests from 1,100 projects hosted on GitHub. We conducted statistical analyses to understand the influence

of these factors on each project individually, and then examined how they varied from one project to another. Additionally, we grouped the projects based on influential factors and identified common trends across the projects. Our findings revealed intriguing insights into pull request merge decisions. We discovered that certain factors have entirely different influences on merge decisions across different projects. These comprehensive analyses shed light on the intricate nature of pull request merges and provide valuable information for developers and project managers to make informed decisions.

The primary contributions of our study include:

- Our research shows that factors have varying impacts on individual projects compared to their impact across multiple projects. This highlights the importance of adapting decision-making strategies to match the unique attributes of each project.
- We discovered that the same factors may have different directions of influence across different projects. This highlights the importance of carefully analyzing each project's context while making contributions.
- We demonstrate that projects can be categorized based on their correlation with different factors. This categorization can assist researchers and practitioners by providing a structured framework to understand the influence of various factors at the project level.

1.3 Organization

The remainder of the thesis is structured as follows: Chapter 2 outlines our methodology, including details on data collection, processing, and analysis methods. Chapter 3 presents the results of our analysis. In Chapter 4, we discuss the key takeaways from our study and contextualize our findings within related works. Chapter 5 addresses the limitations of our analysis and proposes directions for future research.

Chapter 2

Methodology

This chapter outlines the various components of the methodology employed in our study. We begin by explaining the data selection and data filtration procedures. Subsequently, we detail our experimental configuration and the statistical models used to address our research questions. Figure 2.1 provides a brief overview of our methodology.

2.1 Data Collection

For this study, we required a comprehensive dataset containing a significant number of pull requests from diverse projects, along with various associated factors. There are several methods to obtain such data: 1. Extracting from the GitHub data dump, 2. Utilizing the GitHub API to retrieve pull request information, and 3. Reusing datasets from previous research. After exploring each of these options, we decided to pursue the third option—using datasets from prior research—due to its provision of a well-established and reliable data source. Additionally, reusing datasets from previous research enabled us to compare our findings with existing studies and build upon their insights.

After further exploration, we chose to use the dataset [1] curated by Zhang et al. [40]. It offers greater diversity, making it a comprehensive choice for analyzing pull request decisions. Based from this dataset, Zhang et al. studied [41] the overall influence of different factors in pull request decisions. In our study, we first replicated their investigation into the overall influence, and then conducted further analysis to understand project-specific influence.

[1] https://zenodo.org/records/4837135

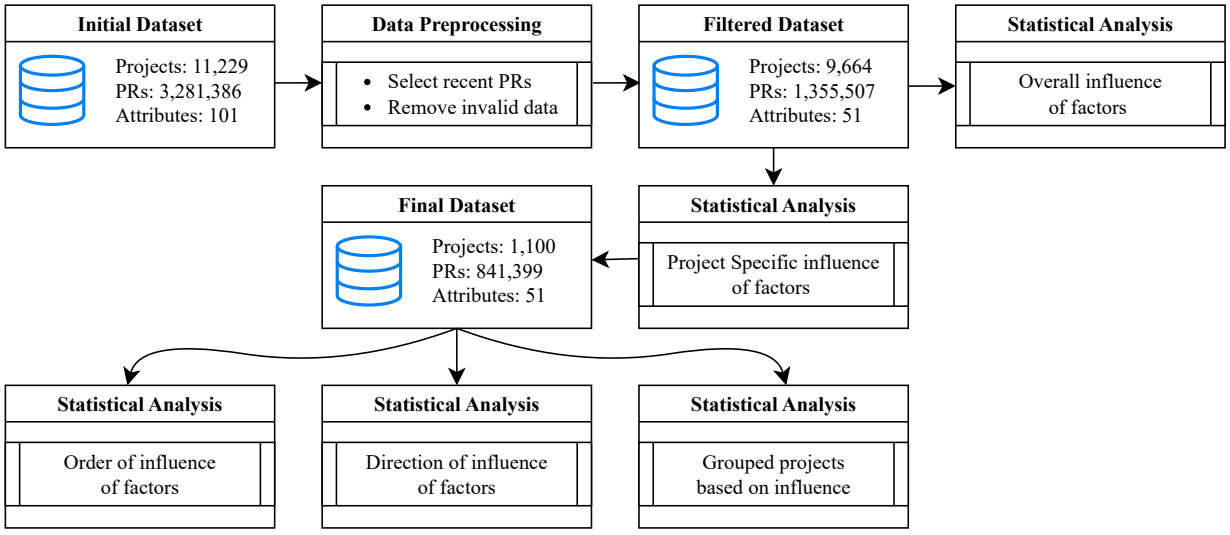

Figure 2.1: An Overview of the Research Methodology

2.2 Data Filtration

The initial dataset contained 3,281,386 pull requests from 11,229 projects, spanning from January 1st, 2011, to September 9, 2018. Each pull request had 101 corresponding attributes, one of which indicated whether it was merged or not. We adopted a pull request filtration process similar to that used in Zhang et al.'s empirical study [41]. Below we briefly describe each step of our filtering process:

- **Pull Request Selection:** To prioritize recent pull requests, we selected only those closed (indicating that a decision had been made) after August 1, 2013. We did this to analyze the most up-to-date and relevant contributions. By excluding pull requests closed before August 1, 2013, we ensured our focus was solely on the latest developments and decisions made by the development team. This approach, previously employed in other studies [40], facilitates a better understanding of the project's current status.

- **Factor Selection:** We specifically chose attributes related to the factors identified in Zeng et al.'s systematic literature review [41] of all factors known to influence pull request decisions. At this stage, each pull request included 51 relevant

attributes, comprising 46 actual factors and the remaining identifiers. Table 2.1 displays the 46 factors, categorized into project, pull request, contributor, integration, and contributor-integrator relationship characteristics.

- **Remove Missing Data:** Finally, we removed the rows with missing data. The final dataset contained 1,355,507 pull requests from 9,664 projects.

Table 2.2 displays our dataset after each stage of filtering.

2.3 Statistical Modeling

To understand the influences of different factors on pull requests, we first preprocessed the data into appropriate forms for our statistical modeling. Then, we adopted a three-fold statistical modeling approach. Firstly, we constructed a generic model to understand the overall influence of factors across all projects. Next, we developed specific models for each individual project to analyze the influence at the project level. Finally, we clustered projects into six distinct groups based on the directions of influence exhibited by different factors. In the following sections, we provide detailed explanations of each of these processes.

2.3.1 Data preprocessing

In this step, first, we converted the string values of the binary factors to numeric values. For instance, the merge status factor had values of 'failure' and 'success', which we changed to 0 and 1 respectively. Then, for continuous variables, we applied a logarithmic transformation, similar to previous studies, [41] and subsequently scaled them. The scaling method calculates the mean and standard deviation of each column, then scales each element by these values—by subtracting the mean and dividing by the standard deviation. This ensures that the mean of each column is 0 and the standard deviation is 1. This helps to make the values more comparable and allows the models to converge better and achieve better accuracy.

2.3.2 Overall influence of factors in pull request decision

We conducted a replication study of Zhang et al. [41] in the context of the overall influence of factors on pull request decisions. In their study, Zhang. et al. used a mixed-effect

Table 2.1: Factors known to influence pull request decision [41]

Factor	Description	Factor	Description
Project Characteristics			
sloc	executable lines of code	open issue num	# of open issues
team size	# of active core team members in last 3 three months	open pr num	# of open pull requests
test lines per kloc	# of test lines per 1K lines of code	pr succ rate	pull request acceptance rate of project
stars	# of stars	pushed delta	# of seconds between two latest pull requests open
project age	# of months from project to pull request creation	integrator availability	latest activity of the two most active integrators
PR Characteristics			
lifetime minutes	# of minutes from pull request creation to latest close time	hash tag	"#" tag exists?
num commits	# of commits	test inclusion	test case existing?
src churn	# of lines changed (added + deleted)	description length	length of pull request description
files added	# of files added	ci exists	uses Continuous Integration?
files deleted	# of files deleted	has comments	pull request has a comment?
files changed	# of files touched	comment conflict	keyword "conflict" exists in comments?
friday effect	pull request submitted on a Friday?	num comments	# of comments
reopen or not	pull request is reopened?	other comment	has noncontributor comment?
commits on files touched	# of commits on files touched	test churn	# of lines of test code changed (added + deleted)
Contributor Characteristics			
contrib open	personality trait: openness	followers	followers at PR creation time
contrib cons	personality trait: conscientious	first pr	first pull request?
contrib extra	personality trait: extraversion	account creation days	days from the contributor's account creation to pull request creation
contrib agree	personality trait: agreeableness	core member	core member?
contrib neur	personality trait: neuroticism	contrib gender	gender? male or female
prev pullreqs	# of previous pull requests		
Integrator Characteristics			
prior review num	# of previous reviews in a project	inte extra	personality trait: extraversion
inte open	personality trait: openness	inte agree	personality trait: agreeableness
inte cons	personality trait: conscientious	inte neur	personality trait: neuroticism
Relationship Characteristic			
same user	same contributor and integrator?		

Table 2.2: Different stages of filtering

Number of	Initial Dataset	Pull Request Filter	Factor Filter	Missing Value FIlter
Project	11,229	11,219	11,219	9,664
Pull Request	3,281,386	1,878,500	1,878,500	1,355,507
Accepted Pull Request	2,765,736	1,583,920	1,583,920	1,180,003
Rejected Pull Request	515,650	294,580	294,580	175,504
Attributes	101	101	51	51

logistic regression model [9] to explore the relationship between each factor and pull request decisions. The project identifier was used as the random effect, denoting similarities among pull requests within a particular project. All other factors were considered to have fixed effects. The glmer function from the lme4 package [8] in R was used to construct the model. The authors used 90% of the data for training the model and the rest for testing. In our study, we first replicated their work and found similar results. Then, we trained the model on the entire dataset. The resulting model represents the overall significance of each factor and the direction of its correlation with pull request decisions (acceptance or rejection).

2.3.3 Project-specific influence of factors in pull request decision

To understand the project-specific influence of factors, we constructed fixed-effect logistic regression models [1] for each project in our dataset. The glm function in R was used to construct the model.

After running fixed-effect models in each of the projects in our dataset, we found that 1,100 out of 9,664 projects have at least one factor that has a statistically significant influence on its pull request merge decision. Table 2.3 displays the summary statistics of

Table 2.3: Summary statistic of number of PRs per project

Statistic	Value
Min	127.0
Q1	363.8
Median	593.0
Mean	988.3
Q3	1101.8
Max	14771.0

the number of PRs per project.

Using these fixed-effect models, we examined how each factor influenced the decision to merge a pull request within its respective project.

2.3.4 Grouping projects based on influence

To group projects with similar influential factors, we used the odds ratio of each factor and employed clustering methods focused on the direction of influence. However, given that not all projects are impacted by every factor, our dataset included missing values. To address this issue, we conducted a partial data cluster analysis using the flipCluster R package from DisplayR. This method extends the k-means clustering approach to address missing values by grouping observations according to shared available data.

2.3.5 Interpretation of statistical models

The influence of different factors and their relative significance on merge decisions can be explained using statistical models with three components: odds ratio, p-value, and percentage variance. The odds ratio quantifies the relationship between a factor and a pull request decision, indicating the increase or decrease in the odds of acceptance for a unit increase in the factor [34]. In this study, a unit of each factor corresponded to one standard deviation from the standardization of the log-transformed factors. The p-value denotes the statistical significance of a factor, signifying the likelihood of evidence against the null hypothesis: suggesting no association between each factor and pull request decisions. Furthermore, the percentage of explained variance serves as a proxy for the relative importance of a factor. This metric, derived from ANOVA Type-II analysis [19], indicates the proportion of variance explained by each factor relative to the total amount of variance, reflecting the effect size in explaining pull request decisions. This measure resembles the percentage of total variance explained by least squares regression [7] and has been used in prior research [27].

2.3.6 Comparison of goodness of fit

Similar to previous studies [26, 41], we evaluated the goodness of fit for each model using the area under the receiver operating characteristic curve (AUC) value. As shown in Figure 2.2, the fit of the fixed-effect models built on individual projects are generally better than that of the mixed-effect model built on the entire dataset.

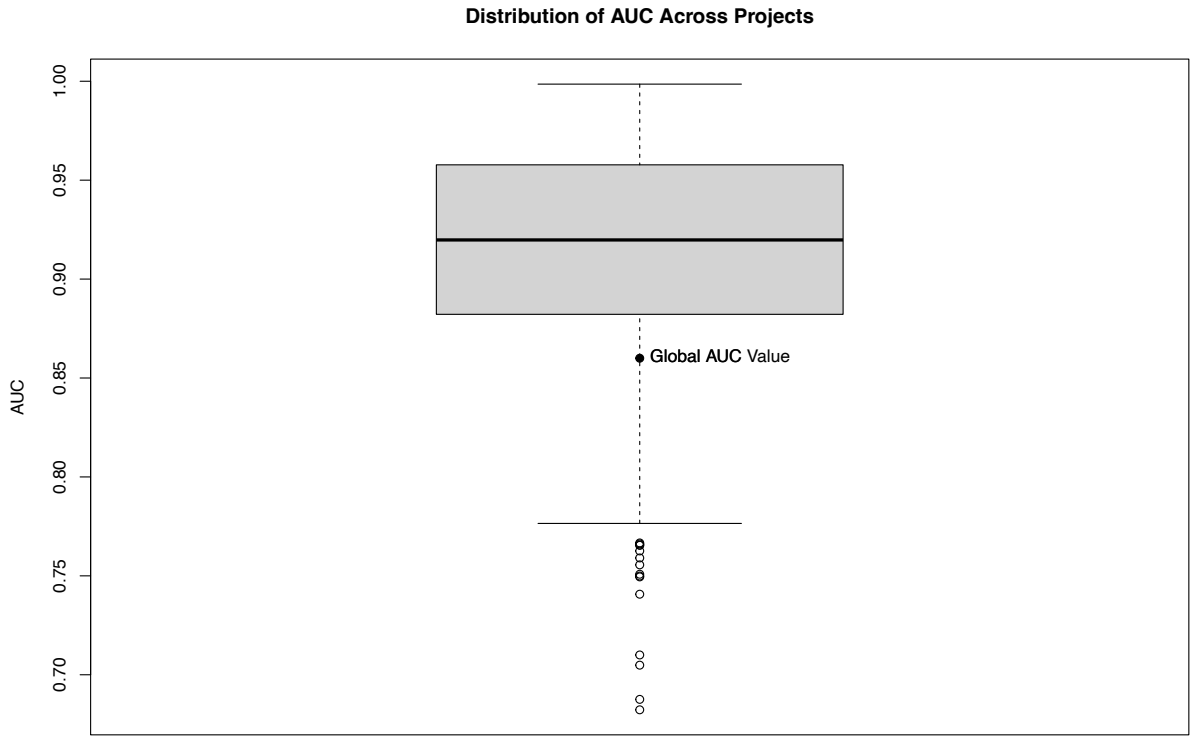

Figure 2.2: Distribution of AUC values

Chapter 3

Results

3.1 RQ1: How does the order of influences of factors differ across projects and within projects?

Motivation:

From a developer perspective, understanding the most influential factors that impact pull request merge decisions in the specific projects they contribute to is beneficial for efficient contribution. Many studies have analyzed pull requests across multiple projects to provide a generalized order of influence across projects. However, this generalized order may not hold for every project, and developers need to know the specific factors that hold the most weight within their project context. This knowledge would allow them to focus their efforts on addressing these key factors and improving their pull request acceptance rates.

Approach:

To understand the difference in the order of influences across projects and within projects, we used the following approach:

First, we built fixed-effects logistic regression models for each of the projects in our dataset. The resulting models indicated the significance of a factor and the direction of its association with a pull request decision (accept or reject) for that specific project. We used the glm function in R to model pull request decisions. The glm function takes the parameter in the form of *response* $\sim$ *terms*, where the *response* is the outcome vector and

terms is the series of independent variables known to be predictors for the response. In our case, the response variable was the outcome of a pull request (accept or reject), and the terms are the dependent factors from our dataset. Then, to get the relative influence,

Then, we used the percentage of explained variance as a substitute for the relative influence of a factor. The variance explained by each factor is calculated from the ANOVA Type-II analysis [19]. When compared to the total variance, the percentage of explained variance acts as a way to estimate the effect size. It helps us understand how much influence a single factor has on explaining pull request decisions. Based on the percentage of explained variance, for each project, we ranked the factors in order of their importance in determining the outcome of a pull request. This ranking allows us to understand which factors have the most significant influence on the merge decision for a specific project.

Finally, we applied the Scott-Knott Effect Size Difference (ESD) test [30] to divide the pull request influencing factors into statistically distinct groups based on the ranks of each factor in each project. The Scott-Knott test uses hierarchical cluster analysis to categorize the classification techniques into ranks. Initially, it divides the classification techniques into two ranks based on the mean AUC values. If these divided ranks exhibit statistically significant differences, the Scott-Knott test further divides them recursively within each ranks. The process continues until the ranks can no longer be subdivided into statistically distinct categories.

Results:

After running fixed-effect models in each of the projects in our dataset, we found that 1,100 out of 9,664 projects have at least one factor that has a statistically significant influence on its pull request merge decision. Figure 3.1 shows the number of projects in which each factor has a statistically significant influence.

Then, we used the Scott-Knott Effect Size Difference (ESD) test to group the factors based on their ranks in different projects. Figure 3.2 displays the result of the Scott-Knott ESD test. Here, the y-axis represents the mean rank of each factor, and the test's identified groups are indicated by different colors. As we can see, there are five distinct groups of factors based on their mean ranks within projects. This implies that the order of influence of these groups varies, with lower mean ranks indicating greater influence on the decision to merge a pull request.

In a previous study, Zhang et al. [41] ranked the factors based on their influence across projects, where they found that the factors *same_ user*, *lifetime_ minutes*, *prior_ review_ num*, *has_ comments*, and *core_ member* tend to be the five most influential

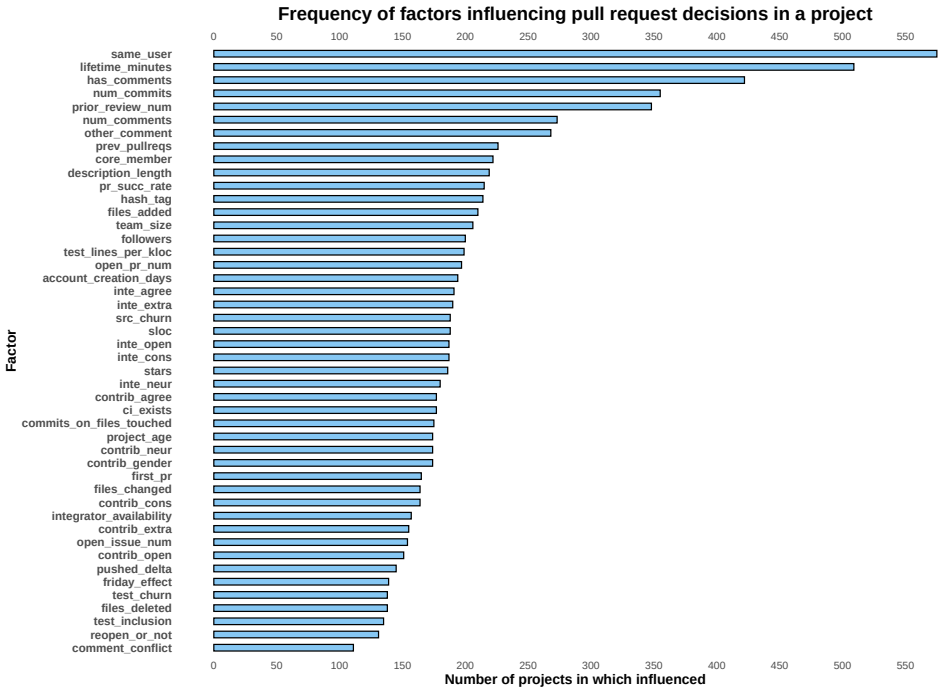

Figure 3.1: Number of projects influenced by each factor

factors in pull request merge decisions. Our results indicate similar findings for the first four factors, as they belong to the top two groups in the Scott-Knott ESD test.

However, despite being ranked as the fifth most influential factor overall across projects, the factor *core_member* does not fall within the first three groups when considering its influence in project-specific contexts. This suggests that while the factor *core_member* may have a significant overall influence on pull request merge decisions, its impact can vary depending on the specific project.

Conversely, *num_comments* was ranked 35th among the 46 factors, suggesting a relatively lower impact when considering overall influence across projects. However, our analysis reveals that it belongs to the fourth group according to the Scott-Knott ESD test, with the lowest mean rank within that group. This indicates a sizeable influence of *num_comments* when project-specific factors are taken into account.

Similarly, the factor *test_lines_per_kloc* was ranked 46th out of the 46 factors in overall influence across projects and did not have a statistically significant influence on merge decisions. However, in our study, we found that this factor had a statistically

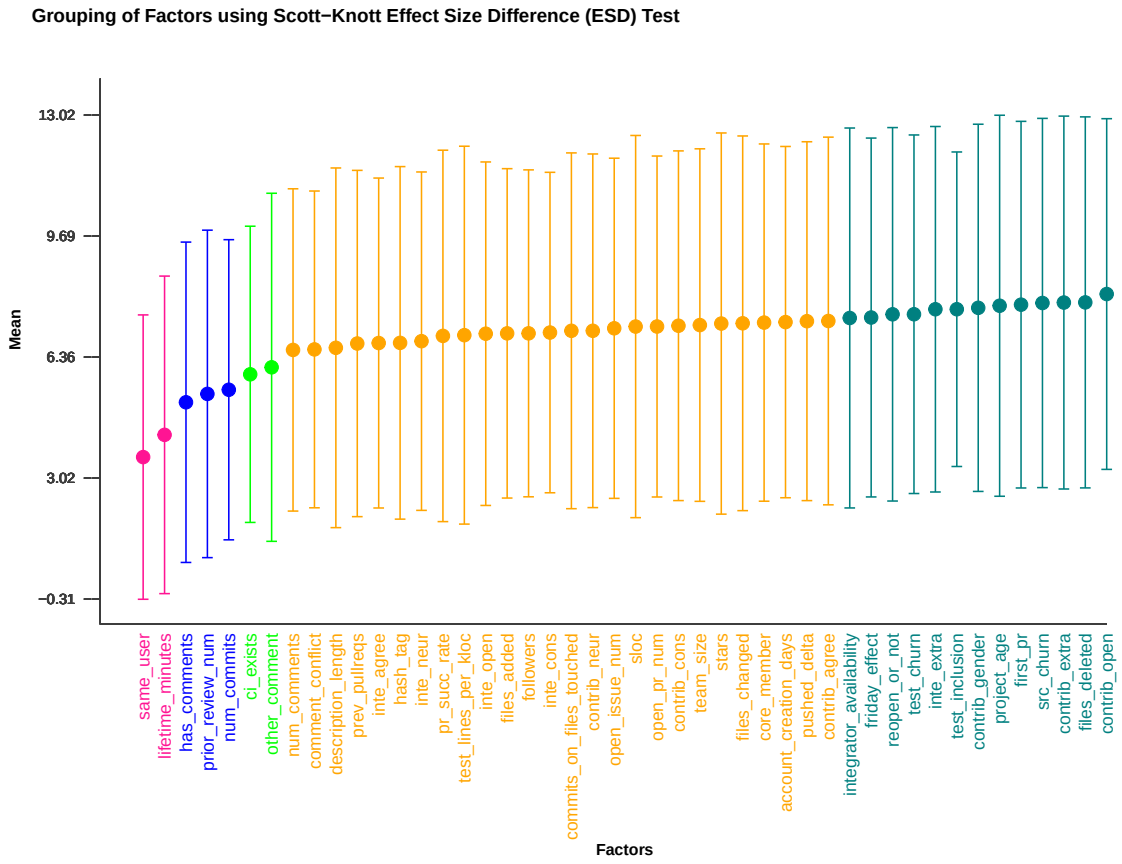

Figure 3.2: Group of factors based on Scott-Knott ESD Test.

significant influence on 199 projects, and the Scott-Knott ESD test placed it in the fourth group of factors out of five. This indicates that while it may not hold a significant influence across all projects, there are instances where it plays an influential role. For developers of those 199 projects, relying solely on the overall influence of this factor and neglecting its importance in creating pull requests may negatively impact their merge decisions.

The order of influences of factors varies significantly across projects and within projects. While certain factors exhibit high overall influence across a broad range of projects, their impact within a specific project context differs.

3.2 RQ2: Do the same factors have different directions of influence in different projects?

Motivation:

A factor's correlation with the merge decision may not ensure consistent direction of influence across projects. For example, including tests may have different influence in different projects. In one project, having test code may increase the likelihood of the pull request being accepted, as tests are seen as a positive contribution by the project maintainers. In another project, tests may be viewed as unnecessary or burdensome for various reasons, resulting in negative outcome of pull requests. Therefore, it is beneficial for a developer to know the direction of influence in the specific project they contribute to, so they can make their pull requests accordingly.

Approach:

To analyze the difference in directions of influence in different projects, we studied the odds ratios of each factor within projects. Odds ratios are a statistical measure used to quantify the direction of association between two variables. In the context of our study, the odds ratio of a factor indicates the likelihood of a particular outcome (in our case acceptance of a pull request) occurring when that factor is present compared to when it is not present. A value greater than 1 suggests that the factor is associated with an increased likelihood of the outcome, indicating a positive direction of influence. Conversely, a value less than 1 suggests that the factor is associated with a decreased likelihood of the outcome, indicating a negative direction of influence.

Results:

We captured the odds ratios for all the factors for each of the projects from our fixed-effect logistic regression models. Figure 3.3 shows the number of projects in each direction of influence, and Figure 3.4 shows the distribution of odds ratios for each factor in multiple projects. Each factor is represented by a box, where the central line inside the box represents the median odds ratio value. The box itself spans the interquartile range (IQR), indicating where the middle 50% of the data lies. The whiskers extending from the boxes show the range of the data, excluding outliers.

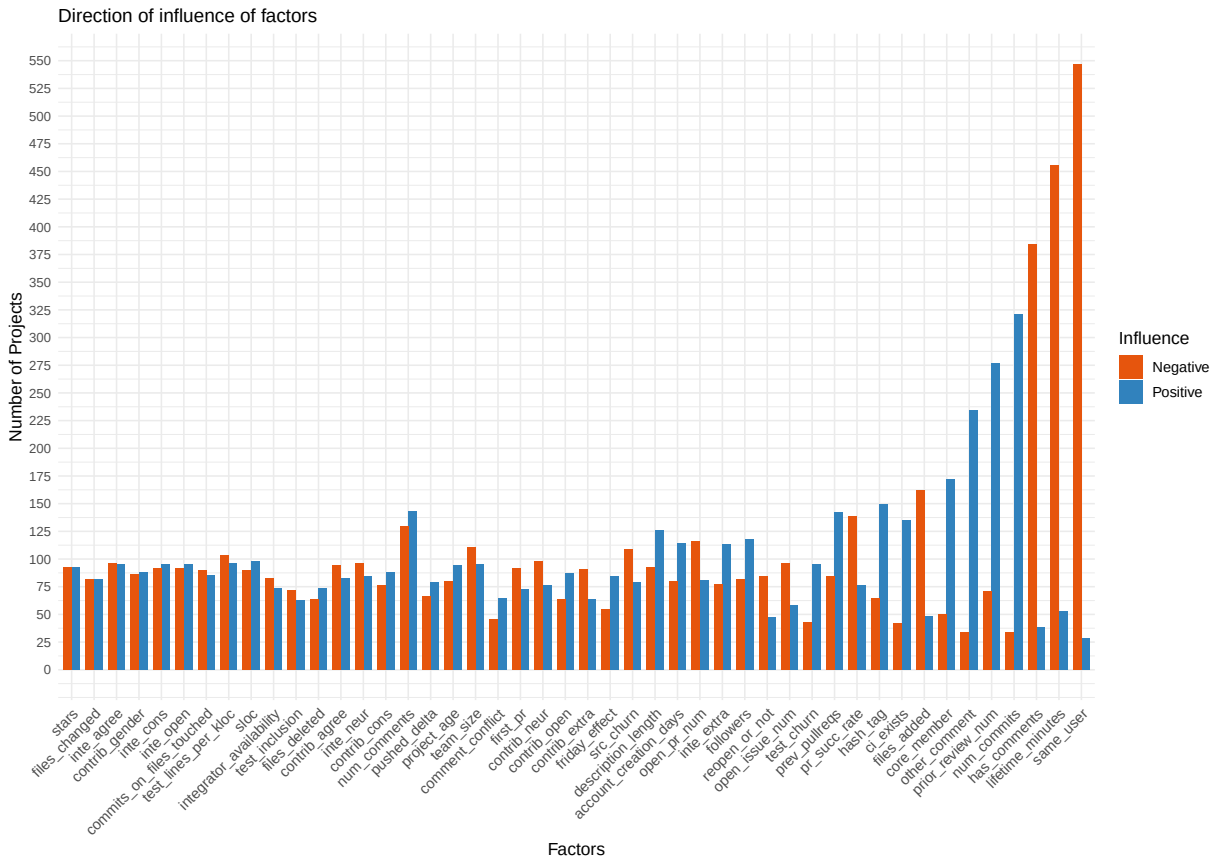

Figure 3.3: Variability in influence across projects

Additionally, there is a horizontal line drawn at y = 1, serving as a reference point to distinguish between positive and negative influence. Factors with odds ratios above this line indicate positive influence, while those below it indicate negative influence. To ensure the readability of the plot, the positive extreme values are scaled within the plot to keep them within a range of 10.

As shown in Figure 3.4, the influences of factors can be categorized into three main groups:

1. **Factors that have mostly positive influence:**
   These factors include *core_ member*, *ci_ exists*, *prior_ review_ number*, *other_ comment*, and *num_ commits*.

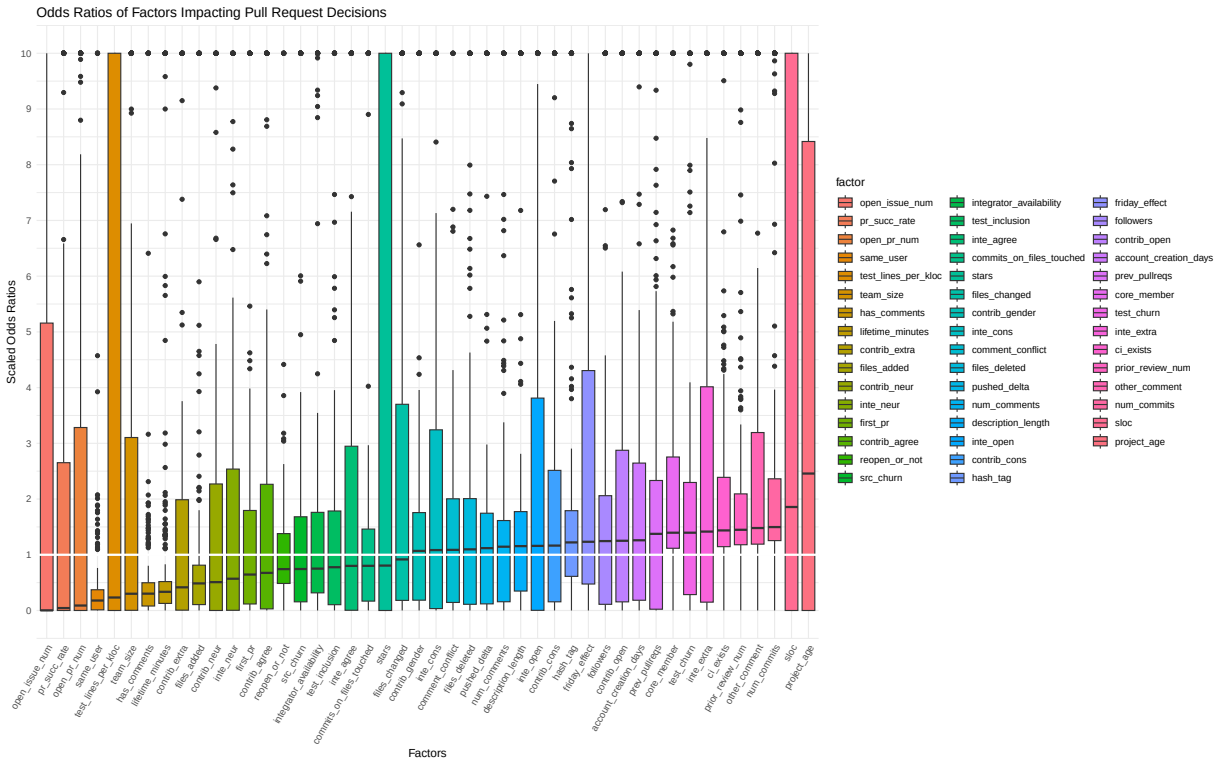

Figure 3.4: Scaled odds-ratios of factors within Project

2. **Factors that have mostly negative influence:**
   These factors include *same_ user*, *has_ comments*, *lifetime_ minutes*, and *files_ added*.

3. **Factors that have mixed influence:**
   Apart from the factors predominantly positive or negative in influence, the remaining factors exhibit mixed influences on merge decisions within project-specific contexts.

While there are a few factors where the direction of influence is consistent across different projects, for most factors, this direction varies. This variation emphasizes the importance of considering project-specific contexts when assessing the influence of a factor.

3.3 RQ3: Which projects are similar in terms of the factors influencing their pull request merge decisions?

Motivation:

Finding similarities in projects based on the factors influencing their merge decisions can offer valuable insights into common patterns and trends within that specific group of projects. This allows for the identification of recurring themes impacting merge outcomes in multiple projects. By studying these themes, researchers can gain a deeper understanding of the dynamics of different projects at the group level.

Approach:

To identify similar projects based on influential factors, we used the odds ratio of each factor and applied clustering techniques based on the direction of influence. However, since not all projects are influenced by every factor, our dataset contained missing values. To address this, we performed a partial data cluster analysis using the flipCluster R package[1] from DisplayR. This clustering technique is an extension of the k-means clustering method that handles missing values by grouping observations based on the data that they have in common.

To determine the optimal value for the number of clusters, we used the Total Sum of Squares (TSS) metric [14]. Starting with $k = 2$, we computed the TSS and iteratively increased the value of k. As we increased k, the TSS continued to rise, indicating improved clustering performance and better representation of the data's underlying structure. However, we observed that beyond $k = 6$, the TSS became undefined, suggesting that further partitioning of the data into clusters was not feasible or meaningful. Therefore, we concluded that $k = 6$ was the optimal value for the number of clusters in our analysis.

Results:

Figure 3.5 illustrates the variation in the direction of influence of different factors across various clusters. The column headers are labeled as clusters 1 to 6, with the numbers in parentheses denoting the number of projects within each cluster. For example, cluster 3 comprises 176 projects. Each cell in the table represents the percentage of projects

[1] https://github.com/Displayr/flipCluster

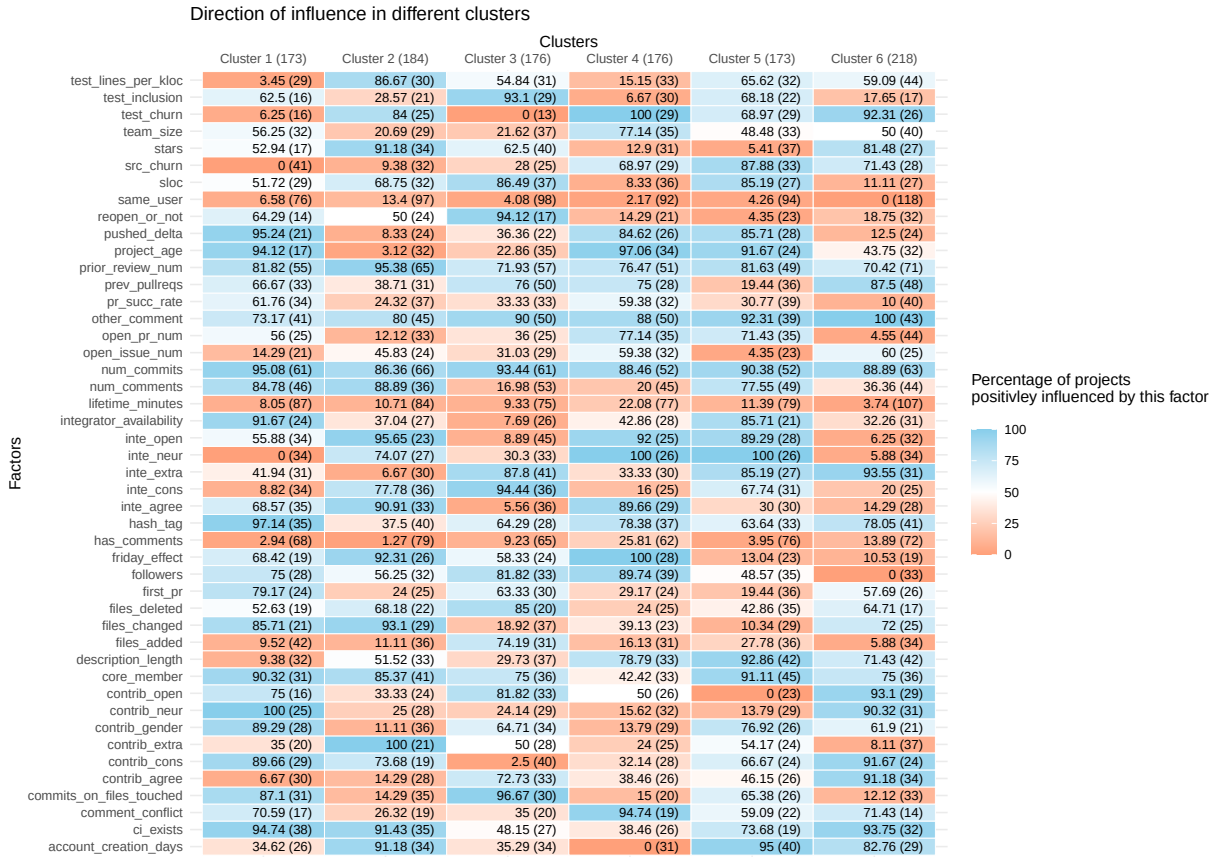

Figure 3.5: Group of factors based on Scott-Knott ESD Test.

positively influenced by the corresponding factor listed on the left. Blue cells indicate a high percentage of positively influenced projects, while red cells indicate a low percentage, implying a higher proportion of negatively influenced projects. The numbers in parentheses within each cell denote the number of projects influenced by that factor within the specific cluster. For instance, in cluster 1, which has a total of 173 projects, 29 projects have been statistically significantly influenced by the factor "*test_lines_per_kloc*" (number of test lines per 1K lines of code), resulting in a positivity rate of 3.45%.

In Figure 3.5, we observe that the same factors show different directions of influence across different project clusters. For example, consider the third factor from the top, '*test_churn*.' This factor represents the number of lines of test code changed (added or deleted) in a pull request. In cluster 4, all 29 projects are positively influenced by this factor, indicating that as the number of lines of test code changed increases, the

likelihood of the pull request being accepted also increases. This suggests that, in this cluster of projects, test code changes are seen as a positive contribution to the pull request, possibly indicating a focus on code quality and stability. Conversely, in cluster 3, this factor negatively influences all 13 projects. Which means, test code changes may be viewed as unnecessary or burdensome in this cluster of projects, leading to a decrease in the likelihood of acceptance.

From a developer perspective, understanding this difference in direction of influence is crucial for determining how to approach making test code changes in their projects. By recognizing whether their project falls into cluster 4 or cluster 3, developers can tailor their contributions accordingly to increase the chances of their pull requests being accepted. This insight highlights the significance of understanding the distinct characteristics of each project when a developer makes contributions.

From our cluster analysis, we identified six groups of projects and observed similarities within each group. For example, projects that are positively correlated with the number of lines changed in pull requests also exhibit a positive correlation with the length of the description.

Chapter 4

Discussion and Related Work

In this section, we will discuss the variations of the influence of all factors across different project clusters, while also exploring relevant literature associated with each factor.

4.1 Factors related to relationship characteristics:

1. **The contributor and integrator being the same person:**

   Zheng et al. [41] first introduced this factor in their study, where they explored the relationship among different factors and empirically explained pull request decisions. They identified this factor as the most influential among all others, noting its negative influence on merge decisions.

   In our study, we observed similar findings. This factor had the lowest mean ranking of all the factors and belongs to the first group out of five (Figure 3.2), according to the Scott-Knott ESD test. It is also one of the four factors that consistently displayed low odds (Figure 3.4), with its entire box positioned below an odds ratio of 1. The cluster analysis further supports these results (Figure 3.5). Across all clusters, the percentage of projects negatively influenced by this factor is more than 80%, reaching 100% in cluster 6.

   The observation that self-integrated pull requests have a negative influence on merge decisions may initially appear counterintuitive, as one might assume that a developer would be unlikely to reject their own pull request. However, previous research [21] has shown that self-approved contributions tend to be more bug-prone, which could explain why self-rejected pull requests occur.

For instance, a developer may unintentionally overlook potential edge cases or bugs in their code due to mostly focusing on the happy path. In such scenarios, rejecting their own pull request allows them to step back, reassess their work, and potentially identify and rectify these issues before merging the changes. Consequently, a robust code review process and practices like pair programming often prove beneficial in collaborative software engineering.

4.2 Factors related to pull request characteristics:

1. **Lifetime of the pull request:**

   A previous study [41] found this factor to have a strong correlation with merge decisions across various projects, and our findings within individual projects corroborate this observation. This factor had the second-lowest mean ranking of all the factors and belongs to the first group out of five (Figure 3.2), according to the Scott-Knott ESD test. It is also one of only four factors consistently showing low odds (Figure 3.4), with its entire box positioned below an odds ratio of 1. The findings from the cluster analysis also supports these results (Figure 3.5). In every cluster, over 75% of projects are adversely affected by this factor.

   This finding suggests that the longer a pull request remains open, the less likely it is to be accepted. For developers, this insight holds significant importance. The key takeaway here is for developers to maintain communication with potential integrators even after submitting a pull request, actively seeking feedback or merge decisions. Through proactive engagement with integrators and timely resolution of any concerns or feedback, developers can enhance the chances of their pull requests being accepted in a timely manner.

2. **Comments:**

   Previous studies found that the presence of comments (*has_comments*) in pull requests has a negative influence on their merge decisions [29, 10, 41], and as the number of comments (*num_comments*) increases, the likelihood of a pull request being accepted decreases [34, 38, 26, 15].

   At the project-specific level, we found similar results for the factor *has_comments*, as it is among the four factors consistently demonstrating low odds (Figure 3.4), with its entire box situated below an odds ratio of 1. The results from the cluster analysis further validate these findings (Figure 3.5). Across all clusters, more than 70% of projects are adversely impacted by this factor.

However, we found that the influence of *num_comments* varies from project to project (Figure 3.4). For some projects it has a positive influence and for others it has the opposite. One possible implication of this finding is that pull requests requiring further discussion may reflect a higher level of scrutiny from the maintainers. Although this could result in initial rejections, continued discussion and subsequent improvements may eventually lead to a positive outcome.

A potential mitigation strategy could involve initiating the actual coding only after there is clarity on the need and expectations from the contribution, and both the contributor and maintainers are in sync about the requirements. This approach can help minimize further discussions after the pull request is submitted, leading to a smoother merge process. By ensuring alignment on requirements and expectations upfront, developers can reduce the likelihood of rejections and streamline the overall pull request workflow.

3. **Number of commits:**

   The relationship between the number of commits and merge decisions is extensively studied in software engineering literature. Some studies [28, 29, 17] suggests a negative correlation, indicating that a higher number of commits decrease the likelihood of successful merges. However, other research [41, 38] suggests a positive correlation.

   In our analysis, we found the number of commits as an influential factor in pull request merge decisions (Figure 3.2), showing a positive influence on merge outcomes at the project level. However, as noted by Zhang et al. [41], the number of commits may not solely represent the contribution size, which generally has a negative influence [28]. During the review process, contributors often make additional commits to incorporate feedback. Thus, while the number of commits may increase by the time the pull request is closed, it maintains a positive correlation with merge outcomes, aligning with our findings.

4. **Number of lines changed:**

   Several studies [15] [34] [26] [18] found that the number of lines changed in a pull request has a negative influence on merge outcome. However, according to our findings, the direction of the influence varies from project to project (Figure 3.4).

   In our cluster analysis, we identified three out of six clusters, where most of the projects are positively influenced by this factor, as shown in Figure 3.5. Notably, one cluster showed a significantly high positive influence, with nearly 90% of projects having a positive correlation with this factor. Interestingly, within that specific cluster, over 90% of projects also showed a positive correlation with longer descriptions

of pull requests. One possible explanation for this observation is that the integrators maintaining these projects are likely to accept large pull requests, especially when accompanied by detailed descriptions from contributors.

5. **Addition, removal, and modification of files:**

   Similar to a previous study that focused on the overall influence [41], our analysis indicated that factors related to the number of files were not particularly influential within individual projects (Figure 3.2). In terms of the direction of influence, both the *number of files deleted* and *the number of files changed* showed mixed correlation across projects. Our cluster analysis further confirmed this variability for *number of files changed*, with two clusters showing a strong positive correlation, while two other clusters showed a strong negative correlation. This suggests that the influence of this factor varies widely from project to project, highlighting the importance for developers to closely consider the specific impact this factor may have on the projects they contribute to.

   Conversely, the direction of influence of the number of files added is more consistent, as it is one of the four factors that mostly had a negative influence (Figure 3.4), indicating that the chance of acceptance decreases as the number of files added increases. This implies that although the number of files added is not among the most influential factors in many projects, but for the small number of projects where it has a statistically significant influence, developers should be cautious about making changes to a lot of files, as it may reduce their chance of acceptance.

6. **Test code:**

   While prior research [15] [38] [26] [34] generally indicates a positive correlation between including tests and pull request acceptance, our analysis at the project level observed wide variability (Figure 3.4) across different projects.

   In our cluster analysis (Figure 3.5), we found one cluster where more than 90% of projects are positively influenced by the *presence of test code* in pull requests, while another cluster showed more than 90% of projects being negatively influenced by the same factor. Similar variability was observed for *the number of lines changed in test code*, with one cluster showing a positive correlation for 100% of projects and another cluster showing a negative correlation for 100% of projects.

   Our findings suggest that the influence of test code cannot be generalized across projects, as it highly depends on the context of each project.

7. **Length of the description:**

In a previous study, Yu et al. [38] analyzed 103,284 pull requests from 40 projects and discovered a negative correlation between the length of the description of pull requests and the merge outcome. However, in our analysis at the project level, we observed its influence to vary from project to project (Figure 3.5). In one cluster of projects, more than 90% of the projects were negatively influenced by this factor. However, in three other clusters, most of the projects were positively influenced.

This finding suggests that the relationship between the pull request description and its impact on the merge outcome may not be generalized. Some integrators prefer detailed descriptions to better understand the changes being made, while others may prioritize brevity and focus more on the code itself. Ultimately, the impact of the pull request description on the merge outcome may depend on the individual preferences and practices within each project team.

8. **Continuous Integration:**

   Similar to previous studies [39, 38], we found that continuous integration (CI) is one of the top influencing factors in pull request merge decisions at the project level. While most factors have mixed influence in different projects, the presence of a CI tool is one of the five factors that mostly have a positive influence.

4.3 Factors related to contributor characteristics:

1. **Contributor being a core member:**

   According to previous studies [41, 15, 38, 34, 6, 20], being a core member has a significant influence on pull request merge decisions and has a positive correlation with the decision to merge. In our study at the project level, we found similar results in terms of the correlation. This factor belongs to a group of five factors where there is a consistent and statistically significant increase in the likelihood of a successful merge when the contributor is a core member (Figure 3.4).

   The positive correlation is understandable, as core members typically serve as project maintainers who possess expertise in the project and are familiar with its intricate details. One key takeaway from this finding for newer contributors is to study the contributions of the core members, as this allows them to understand how the project evolved over time and what strategies were successful in contributing to its success. By analyzing the work of core members, newer contributors can gain valuable insights into the project's objectives, priorities, and best practices. This knowledge can help

them align their own contributions with the project's overall direction and may lead to more successful pull requests.

2. **Experience of the contributor:**

   Previous studies found that, generally, the number of past pull requests from a contributor [18, 26] and the number of followers a contributor has [34, 38, 26] are positively correlated with merge outcome. At the same time, being a first-time contributor (*first_pr*) [20] or having a relatively new account (*account_creation_days*) [26] are negatively correlated with merge outcome.

   This suggests experience is valued in the open source community. As contributors gain more experience, the likelihood of their pull requests being accepted increases. This is understandable since experienced contributors tend to be more familiar with the project's guidelines, coding conventions, and community norms, which can lead to higher-quality contributions that are more likely to be accepted.

   In our study at the project level, we also found that most projects have a positive correlation with contributors' experience, but there are exceptions (Figure 3.4). In one of our six clusters of projects, the majority of projects did not display a positive correlation with the number of past pull requests from the contributor. This may suggest that while having previous experience may facilitate the merging of pull requests, there are projects where contributions from less experienced developers are still welcomed. From a new contributor's perspective, being able to identify newcomer-friendly projects can be highly beneficial.

3. **Personality traits:** Similar to the previous study by Zhang et al. [41], we did not find the personality trait of contributors among the most influential factors in merge outcome (Figure 3.2). In terms of directionality, Iyer et al. [15] discovered that contributors' openness and conscientiousness positively influence pull request acceptance, whereas extraversion has a negative impact. Additionally, agreeableness and neuroticism were found to have statistically insignificant influence. While analyzing personality traits at a project level, we found their influence to vary from project to project (Figure 3.4). This finding suggests that the influence of human factors in pull request merge decisions is context-dependent and may not be generalized.

4.4 Factors related to integrator characteristics:

1. **Experience of the integrator:**

In previous research, there are mixed findings regarding this factor. Baysal et al. [5] did not find a significant association between the experience of the integrator and pull request decisions. However, Zhang et al. [41] found this factor to be significantly important, ranking it third in overall influence across various projects and having an overall positive influence on merge decisions. It is important to note that Baysal et al.'s research was a case study of two projects, while Zhang et al.'s study consists of thousands of projects. This explains the variability in their results.

Our results are more aligned with the studies of Zhang et al. We found the experience of the integrator to have a significant correlation with merge outcomes (Figure 3.2). Additionally, it is one of the five factors consistently demonstrating high odds (Figure 3.4), with its entire box situated above an odds ratio of 1.

2. **Personality traits:** Similar to the personality traits of contributors, Zhang er al. [41] did not find the personality trait of integrators among the most influential factors in merge outcome, which aligns with our finding (Figure 3.2). In terms of directionality, Iyer et al. [15] found all five personality traits of integrators (openness, conscientiousness, extraversion, agreeableness, neuroticism) to positively influence pull request acceptance. Our observations at the project level suggest that, like contributors, the personality traits of integrators also vary across projects (Figure 3.4).

4.5 Factors related to project characteristics:

A recent study [40] analyzing pull requests from 11,230 projects found that factors associated with project characteristics contribute to only approximately 1% of the variance, suggesting their limited influence on pull request decisions. We found similar results in our study, considering within project-specific context. None of the project-related factors ranked among the top three groups in our Scott-Knott ESD test.

However, we found an interesting insight for the factors *project_age* and *stars*. Previous studies [15, 26, 34] found that, as projects age and the number of stars increases, the likelihood of a successful merge decreases. This trend may discourage new contributors from engaging in mature and popular open-source projects. However, in our cluster analysis, we found one cluster where more than 90% of projects are positively correlated with the factor project_age and around 80% of the projects are also positively correlated with the fact that this was the first pull request of the contributor. This indicates that, despite being less common, there are old and mature projects that are very welcoming to

new contributors, and for a developer who wants to start contributing to well-established open-source projects, selecting the right project is crucial.

For the other project related factors (*team_size, test_lines per_kloc, open_issue_num, open_pr_num, pr_succ_rate, pushed_delta, and integrator_availability*), we observed wide variability across different projects, suggesting the influence of project characteristics may not be generalized.

Chapter 5

Conclusions:

5.1 Limitations and Future Work

While we considered a comprehensive list of factors known to influence merge outcomes in previous studies, there are additional factors that were not included in this study. One such factor is the code itself, specifically how well it serves the purpose of the pull request in question. Understanding the effectiveness of a pull request requires a thorough understanding of several factors such as the project's domain, coding standards, and best practices, which vary widely across different projects. Additionally, quantifying the effectiveness of the code often involves subjective assessments that may vary from one integrator to another. For these reasons, the specifics of the code being merged were kept out of scope in this study.

In this study, we did not conduct correlation analysis at the project level. This implies that there might still be relationships between factors within individual projects that weren't captured in our research. However, for the dataset we used [40], the authors performed a comprehensive correlation analysis and excluded many factors due to their strong correlations with others [41]. The selection of factors was influenced by their use in previous studies, frequency in the literature, promising performance, expressiveness, and data availability. For example, from the correlated factors *test_lines_per_kloc, test_cases_per_kloc, and asserts_per_kloc*, the factor *test_lines_per_kloc* was selected based on previous study [11].

The relative order of influence of factors may change depending on the selection of different methods. There are several ways to calculate the importance of factors in a

logistic regression model, such as the percentage of variance explained by each factor [27, 7] and the standardized coefficient [33]. In our study, we chose the percentage of explained variance to measure the influence of factors, a widely used metric in many related works [41, 38, 22].

Based on the findings of our study, we aim to develop a tool integrated directly with GitHub to assist developers in customizing their pull requests according to specific project requirements. By leveraging this tool, developers will gain insights into the influential factors affecting merge decisions within a particular project. They can then adjust their contributions accordingly, leading to improved efficiency in projects using the pull-based development model.

5.2 Contributions

In this study, we analyzed 1,100 open-source projects to explore the impact of various factors on merge decisions at the project level. We built regression models for each project and found that both the order of influence of factors and their directions vary from project to project. Our findings have implications for both researchers and practitioners.

For contributors, our research highlights the significance of recognizing the unique context of each project. By tailoring pull requests to align with project-specific requirements, contributors can enhance the likelihood of successful merges.

For integrators, our findings help better understand how their merging decisions are correlated with different factors and improve their merging strategies by focusing on more important factors if required.

For researchers, these findings present an opportunity to investigate further into the factors influencing merge decisions at the project level. By scrutinizing these influences, researchers can gain insights into why certain factors vary in their impact across different projects. This investigation can lead to a better understanding of the complexities of software development processes and contribute to advancements in the field of software engineering research.

5.3 Conclusions

In this study, we examined 46 factors and evaluated their influence on pull request merge decisions at the project level. We analyzed over 10,000 projects using fixed-effect logistic

regression models and identified 1,100 projects that exhibit at least one influencing factor. Our results show that the order and direction of influence of factors vary greatly from project to project. For example, factors such as the number of files changed, test lines, comments, age of the project, and team size have completely different impacts in different projects. In half of the projects we studied, these factors had a positive influence, while the opposite was observed in the others. Additionally, we found that a factor may have a crucial influence in one project but not in others.

Our findings indicate that the dynamics of pull requests within each project are closely related to the context, and a microscopic view at the project level is required to understand how these factors influence merge decisions. It is important to analyze each project individually to determine which factors are most influential and how they interact with one another. By closely examining the specific characteristics of each project, developers can better understand how to manage and optimize pull request dynamics to make effective and efficient contributions. Ultimately, these findings suggest that a one-size-fits-all approach to pull request management may not be effective, and instead, a tailored strategy based on the unique context of each project is necessary.

References

[1] Paul D Allison. *Fixed effects regression models*. SAGE publications, 2009.

[2] Douglas M Bates. lme4: Mixed-effects modeling with r, 2010.

[3] Olga Baysal, Oleksii Kononenko, Reid Holmes, and Michael W Godfrey. The secret life of patches: A firefox case study. In *2012 19th working conference on reverse engineering*, pages 447–455. IEEE, 2012.

[4] Olga Baysal, Oleksii Kononenko, Reid Holmes, and Michael W Godfrey. The influence of non-technical factors on code review. In *2013 20th working conference on reverse engineering (WCRE)*, pages 122–131. IEEE, 2013.

[5] Olga Baysal, Oleksii Kononenko, Reid Holmes, and Michael W Godfrey. Investigating technical and non-technical factors influencing modern code review. *Empirical Software Engineering*, 21:932–959, 2016.

[6] Amiangshu Bosu and Jeffrey C Carver. Impact of developer reputation on code review outcomes in oss projects: An empirical investigation. In *Proceedings of the 8th ACM/IEEE international symposium on empirical software engineering and measurement*, pages 1–10, 2014.

[7] Jacob Cohen, Patricia Cohen, Stephen G West, and Leona S Aiken. *Applied multiple regression/correlation analysis for the behavioral sciences*. Routledge, 2013.

[8] Tapajit Dey and Audris Mockus. Which pull requests get accepted and why? a study of popular npm packages. *arXiv preprint arXiv:2003.01153*, 2020.

[9] Andrzej Gałecki, Tomasz Burzykowski, Andrzej Gałecki, and Tomasz Burzykowski. *Linear mixed-effects model*. Springer, 2013.

[10] Mehdi Golzadeh, Alexandre Decan, and Tom Mens. On the effect of discussions on pull request decisions. In *BENEVOL*, 2019.

[11] Georgios Gousios, Martin Pinzger, and Arie van Deursen. An exploratory study of the pull-based software development model. In *Proceedings of the 36th international conference on software engineering*, pages 345–355, 2014.

[12] Georgios Gousios and Andy Zaidman. A dataset for pull-based development research. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pages 368–371, 2014.

[13] Georgios Gousios, Andy Zaidman, Margaret-Anne Storey, and Arie Van Deursen. Work practices and challenges in pull-based development: The integrator's perspective. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 1, pages 358–368. IEEE, 2015.

[14] Salvatore Ingrassia and Antonio Punzo. Cluster validation for mixtures of regressions via the total sum of squares decomposition. *Journal of Classification*, 37(2):526–547, 2020.

[15] Rahul N Iyer, S Alex Yun, Meiyappan Nagappan, and Jesse Hoey. Effects of personality traits on pull request acceptance. *IEEE Transactions on Software Engineering*, 47(11):2632–2643, 2019.

[16] Yujuan Jiang, Bram Adams, and Daniel M German. Will my patch make it? and how fast? case study on the linux kernel. In *2013 10th Working conference on mining software repositories (MSR)*, pages 101–110. IEEE, 2013.

[17] Nikhil Khadke, Ming Han Teh, and Minghan Shen. Predicting acceptance of github pull requests. *Stanford–CS 229, Tech. Rep*, 2012.

[18] Oleksii Kononenko, Tresa Rose, Olga Baysal, Michael Godfrey, Dennis Theisen, and Bart De Water. Studying pull request merges: a case study of shopify's active merchant. In *Proceedings of the 40th international conference on software engineering: software engineering in practice*, pages 124–133, 2018.

[19] Øyvind Langsrud. Anova for unbalanced data: Use type ii instead of type iii sums of squares. *Statistics and computing*, 13(2):163–167, 2003.

[20] Amanda Lee and Jeffrey C Carver. Are one-time contributors different? a comparison to core and periphery developers in floss repositories. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–10. IEEE, 2017.

[21] Shane McIntosh, Yasutaka Kamei, Bram Adams, and Ahmed E Hassan. The impact of code review coverage and code review participation on software quality: A case study of the qt, vtk, and itk projects. In *Proceedings of the 11th working conference on mining software repositories*, pages 192–201, 2014.

[22] Cassandra Overney, Jens Meinicke, Christian Kästner, and Bogdan Vasilescu. How to not get rich: An empirical study of donations in open source. In *Proceedings of the ACM/IEEE 42nd international conference on software engineering*, pages 1209–1221, 2020.

[23] Rohan Padhye, Senthil Mani, and Vibha Singhal Sinha. A study of external community contribution to open-source projects on github. In *Proceedings of the 11th working conference on mining software repositories*, pages 332–335, 2014.

[24] Gustavo Pinto, Luiz Felipe Dias, and Igor Steinmacher. Who gets a patch accepted first? comparing the contributions of employees and volunteers. In *Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 110–113, 2018.

[25] Mohammad Masudur Rahman and Chanchal K Roy. An insight into the pull requests of github. In *Proceedings of the 11th working conference on mining software repositories*, pages 364–367, 2014.

[26] Ayushi Rastogi, Nachiappan Nagappan, Georgios Gousios, and André van der Hoek. Relationship between geographical location and evaluation of developer contributions in github. In *Proceedings of the 12th ACM/IEEE international symposium on empirical software engineering and measurement*, pages 1–8, 2018.

[27] Baishakhi Ray, Daryl Posnett, Vladimir Filkov, and Premkumar Devanbu. A large scale study of programming languages and code quality in github. In *Proceedings of the 22nd ACM SIGSOFT international symposium on foundations of software engineering*, pages 155–165, 2014.

[28] Daricélio Moreira Soares, Manoel Limeira de Lima Júnior, Leonardo Murta, and Alexandre Plastino. Acceptance factors of pull requests in open-source projects. In

Proceedings of the 30th Annual ACM Symposium on Applied Computing, pages 1541–1546, 2015.

[29] Daricélio Moreira Soares, Manoel L De Lima Junior, Leonardo Murta, and Alexandre Plastino. Rejection factors of pull requests filed by core team developers in software projects with high acceptance rates. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, pages 960–965. IEEE, 2015.

[30] Chakkrit Tantithamthavorn, Shane McIntosh, Ahmed E Hassan, and Kenichi Matsumoto. An empirical comparison of model validation techniques for defect prediction models. *IEEE Transactions on Software Engineering*, 43(1):1–18, 2016.

[31] Yida Tao, Donggyun Han, and Sunghun Kim. Writing acceptable patches: An empirical study of open source project patches. In *2014 IEEE International Conference on Software Maintenance and Evolution*, pages 271–280. IEEE, 2014.

[32] Josh Terrell, Andrew Kofink, Justin Middleton, Clarissa Rainear, Emerson R Murphy-Hill, and Chris Parnin. Gender bias in open source: Pull request acceptance of women versus men. *PeerJ Prepr.*, 4:e1733, 2016.

[33] Scott Tonidandel and James M LeBreton. Relative importance analysis: A useful supplement to regression analysis. *Journal of Business and Psychology*, 26:1–9, 2011.

[34] Jason Tsay, Laura Dabbish, and James Herbsleb. Influence of social and technical factors for evaluating contribution in github. In *Proceedings of the 36th international conference on Software engineering*, pages 356–366, 2014.

[35] Bogdan Vasilescu, Yue Yu, Huaimin Wang, Premkumar Devanbu, and Vladimir Filkov. Quality and productivity outcomes relating to continuous integration in github. In *Proceedings of the 2015 10th joint meeting on foundations of software engineering*, pages 805–816, 2015.

[36] Peter Weißgerber, Daniel Neu, and Stephan Diehl. Small patches get in! In *Proceedings of the 2008 international working conference on Mining software repositories*, pages 67–76, 2008.

[37] Yue Yu, Huaimin Wang, Vladimir Filkov, Premkumar Devanbu, and Bogdan Vasilescu. Wait for it: Determinants of pull request evaluation latency on github. In *2015 IEEE/ACM 12th working conference on mining software repositories*, pages 367–371. IEEE, 2015.

[38] Yue Yu, Gang Yin, Tao Wang, Cheng Yang, and Huaimin Wang. Determinants of pull-based development in the context of continuous integration. *Science China Information Sciences*, 59:1–14, 2016.

[39] Fiorella Zampetti, Gabriele Bavota, Gerardo Canfora, and Massimiliano Di Penta. A study on the interplay between pull request review and continuous integration builds. In *2019 IEEE 26th international conference on software analysis, evolution and reengineering (SANER)*, pages 38–48. IEEE, 2019.

[40] Xunhui Zhang, Ayushi Rastogi, and Yue Yu. On the shoulders of giants: A new dataset for pull-based development research. In *Proceedings of the 17th international conference on mining software repositories*, pages 543–547, 2020.

[41] Xunhui Zhang, Yue Yu, Georgios Gousios, and Ayushi Rastogi. Pull request decisions explained: An empirical overview. *IEEE Transactions on Software Engineering*, 49(2):849–871, 2022.

APPENDICES

Appendix A

List of Projects

Table A.1 lists the 1,100 projects that were found to have at least one influential factor in our analysis.

Table A.1: Repository information of the projects

#	**ownername**	**reponame**
1	stylelint	stylelint
2	binary-com	SmartCharts
3	letsencrypt	boulder
4	candlepin	subscription-manager
5	Zeit	docs
6	karma-runner	karma
7	apache	zeppelin
8	pentaho	pentaho-kettle
9	JetBrains	intellij-community
10	apache	spark
11	GoogleCloudPlatform	google-cloud-eclipse
12	Netflix	conductor

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
13	project-ncl	pnc
14	scikit-learn	scikit-learn
15	mulesoft	mule
16	saltstack	salt
17	opf	openproject
18	rapid7	metasploit-framework
19	akka	akka
20	prestodb	presto
21	AnalyticalGraphicsInc	cesium
22	kubernetes	kubernetes
23	pcgen	pcgen
24	infinispan	infinispan
25	hazelcast	hazelcast
26	zotero	translators
27	triplea-game	triplea
28	Fyrd	caniuse
29	Minio	minio
30	Ramda	ramda
31	facebook	react
32	checkstyle	checkstyle
33	discourse	discourse
34	mit-cml	appinventor-sources
35	OriginTrail	ot-node
36	netlify	netlify-cms

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
37	nodejs	node
38	MichMich	MagicMirror
39	SleepyTrousers	EnderIO
40	kiegroup	droolsjbpm-integration
41	mozilla	pdf.js
42	ManageIQ	manageiq-ui-classic
43	GoCD	GoCD
44	eclipse	californium
45	datastax	python-driver
46	ansible	ansible
47	JMRI	JMRI
48	playframework	playframework
49	cdk	cdk
50	sparc-request	sparc-request
51	obiba	ng-obiba-mica
52	apache	nifi
53	pallets	flask
54	edx	configuration
55	frappe	erpnext
56	odoo	odoo
57	scrapy	scrapy
58	mrdoob	three.js
59	python	peps
60	grafana	grafana

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
61	apache	beam
62	sunpy	sunpy
63	weaveworks	weave
64	Graylog2	graylog2-server
65	dcos	dcos
66	OpenNMS	opennms
67	neo4j	neo4j
68	Cilium	cilium
69	google	blockly
70	openshift	openshift-ansible
71	cfpb	cfgov-refresh
72	openshift	origin
73	control-center	serviced
74	elastic	beats
75	duckduckgo	zeroclickinfo-spice
76	code-dot-org	code-dot-org
77	openstax	tutor-server
78	rails	rails
79	cloudify-cosmo	cloudify-cli
80	resteasy	Resteasy
81	proofpoint	platform
82	payara	Payara
83	matplotlib	matplotlib
84	HubSpot	Singularity

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
85	keycloak	keycloak
86	Yakindu	statecharts
87	DataDog	datadog-agent
88	dimagi	commcare-android
89	plataformatec	devise
90	k9mail	k-9
91	google	ggrc-core
92	theforeman	foreman
93	hail-is	hail
94	sympy	sympy
95	blevesearch	bleve
96	python	mypy
97	apache	ignite
98	khartec	waltz
99	EFForg	https-everywhere
100	SonarSource	sonar-java
101	liberapay	liberapay.com
102	avocado-framework	avocado
103	puma	puma
104	choderalab	yank
105	ORCID	ORCID-Source
106	dimagi	commcare-cloud
107	ethereum	go-ethereum
108	LightningNetwork	lnd

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
109	docker	swarm
110	eslint	eslint
111	Leaflet	Leaflet
112	projectcalico	calico
113	ilios	frontend
114	insolar	insolar
115	OCA	server-tools
116	emberjs	ember.js
117	mendersoftware	meta-mender
118	honestbleeps	Reddit-Enhancement-Suite
119	pandas-dev	pandas
120	vitessio	vitess
121	reactor	reactor-core
122	Activiti	Activiti
123	apache	couchdb-fauxton
124	guardian	subscriptions-frontend
125	antlr	antlr4
126	fastlane	fastlane
127	ENCODE-DCC	encoded
128	mozilla	kitsune
129	github	linguist
130	docker	cli
131	GoogleCloudPlatform	python-docs-samples
132	pantsbuild	pants

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
133	obiba	mica2
134	otwcode	otwarchive
135	Dogfalo	materialize
136	certbot	certbot
137	VoltDB	voltdb
138	ow2-proactive	scheduling
139	SpongePowered	SpongeCommon
140	uber	deck.gl
141	guardian	grid
142	DevExpress	testcafe-hammerhead
143	apache	kafka
144	antirez	redis-doc
145	openstreetmap	openstreetmap-website
146	linode	manager
147	camptocamp	ngeo
148	evergreen-ci	evergreen
149	RaRe-Technologies	gensim
150	samtools	htsjdk
151	asciidoctor	asciidoctor
152	Crate	crate
153	buildbot	buildbot
154	styled-components	styled-components
155	aws	aws-sdk-ruby
156	elastic	eui

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
157	codemirror	CodeMirror
158	OpenLiberty	open-liberty
159	adobe	brackets-shell
160	openSUSE	open-build-service
161	grails	grails-core
162	google	closure-compiler
163	jbosstools	jbosstools-integration-stack-tests
164	developit	preact
165	molgenis	molgenis
166	Alluxio	alluxio
167	esy	esy
168	ember-cli	ember-cli
169	apache	incubator-druid
170	spring-projects	spring-boot
171	newsuk	times-components
172	ExpressGateway	express-gateway
173	influxdata	telegraf
174	jhipster	generator-jhipster
175	opendatakit	collect
176	koajs	koa
177	intel-analytics	BigDL
178	SatelliteQE	robottelo
179	zenoss	zenoss-prodbin
180	loomio	loomio

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
181	aerogear	aerogear-unifiedpush-server
182	improbable-eng	thanos
183	videojs	video.js
184	scipy	scipy
185	convox	rack
186	oat-sa	tao-core
187	babel	babel
188	Sage-Bionetworks	Synapse-Repository-Services
189	Talend	tdi-studio-se
190	inspirehep	inspire-next
191	teiid	teiid
192	TerriaJS	terriajs
193	windup	windup
194	nasa	openmct
195	puppetlabs	puppet
196	eclipse	smarthome
197	prometheus	prometheus
198	travis-ci	travis-build
199	bosun-monitor	bosun
200	FasterXML	jackson-databind
201	cerebris	jsonapi-resources
202	petkaantonov	bluebird
203	18F	federalist
204	juju	juju-gui

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
205	Minio	mc
206	spring-projects	spring-framework
207	go-swagger	go-swagger
208	18F	calc
209	Financial-Times	polyfill-service
210	spring-projects	spring-security
211	alphagov	notifications-admin
212	openshift	openshift-tools
213	department-of-veterans-affairs	caseflow
214	AsyncHttpClient	async-http-client
215	renovatebot	renovate
216	spring-projects	spring-integration
217	broadinstitute	cromwell
218	scala	scala
219	SonarSource	sonarqube
220	sequelize	sequelize
221	OpenCollective	opencollective-api
222	SeleniumHQ	selenium
223	spyder-ide	spyder
224	sbt	sbt-native-packager
225	storybooks	storybook
226	ipython	ipython
227	ipfs	js-ipfs
228	cakephp	docs

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
229	wellcometrust	platform
230	pentaho	pentaho-platform
231	tootsuite	mastodon
232	DIRACGrid	DIRAC
233	wazuh	wazuh-documentation
234	ManageIQ	integration_tests
235	chef	chef-dk
236	broadinstitute	gatk
237	OpenClinica	OpenClinica
238	sitespeedio	sitespeed.io
239	falconry	falcon
240	mysociety	alaveteli
241	Catrobat	Catroid
242	spotify	docker-client
243	jruby	jruby
244	openhab	openhab2-addons
245	sphinx-doc	sphinx
246	test-kitchen	test-kitchen
247	learning-unlimited	ESP-Website
248	hypothesis	h
249	IOTAledger	trinity-wallet
250	edx	edx-platform
251	DHIS2	dhis2-core
252	Nexmo	nexmo-developer

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
253	oppia	oppia
254	deeplearning4j	deeplearning4j
255	uclouvain	OSIS
256	NixOS	nixops
257	Wikia	mobile-wiki
258	gatsbyjs	gatsby
259	ceph	ceph-ansible
260	Kinto	kinto
261	os-autoinst	openQA
262	datastax	nodejs-driver
263	pyca	cryptography
264	OCA	OpenUpgrade
265	chef	chef
266	gramps-project	gramps
267	stripe	stripe-java
268	angular	material
269	apache	calcite
270	pypa	pip
271	wix	react-native-navigation
272	gobuffalo	buffalo
273	binary-com	binary-static
274	python	cpython
275	edx	edx-ora2
276	apache	bookkeeper

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
277	wbond	package_control_channel
278	kubernetes	dashboard
279	OGGM	oggm
280	polyswarm	polyswarm-client
281	JedWatson	react-select
282	tendermint	tendermint
283	square	p2
284	getredash	redash
285	meetup	meetup-web-components
286	pywbem	pywbem
287	apache	camel
288	hortonworks	cloudbreak
289	flutter	flutter-intellij
290	DestinyItemManager	DIM
291	FreeNAS	freenas
292	mapbox	mapbox-gl-js
293	functional-streams-for-scala	fs2
294	dart-lang	site-www
295	PyCQA	pylint
296	PegaSysEng	pantheon
297	BBC	simorgh
298	mattermost	mattermost-mobile
299	openaps	oref0
300	Blazemeter	taurus

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
301	toptal	chewy
302	apache	couchdb
303	ansible	awx
304	howdyai	botkit
305	dropwizard	metrics
306	pypa	virtualenv
307	ezsystems	ezpublish-legacy
308	Turfjs	turf
309	codice	ddf
310	LMFDB	lmfdb
311	Azure	azure-rest-api-specs
312	Zarel	Pokemon-Showdown-Client
313	Jermolene	TiddlyWiki5
314	vega	vega
315	activemerchant	active_merchant
316	WGBH	AAPB2
317	matrix-org	matrix-doc
318	phenotips	phenotips
319	Microsoft	pai
320	spack	spack
321	Azure	azure-sdk-for-node
322	napalm-automation	napalm
323	gem	oq-engine
324	mozilla	addons-server

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
325	OpenMRS	openmrs-core
326	scionproto	scion
327	graknlabs	grakn
328	OpenSlides	OpenSlides
329	Bytom	bytom
330	ericsson	codechecker
331	Ultimaker	Cura
332	enonic	xp
333	mozilla	activity-stream
334	edx	credentials
335	Dask	distributed
336	Microsoft	pxt
337	openshift	ansible-service-broker
338	angular	angular-cli
339	cardstack	cardstack
340	Zarel	Pokemon-Showdown
341	hibernate	hibernate-tools
342	dcos	dcos-ui
343	xwiki	xwiki-platform
344	ganga-devs	ganga
345	OpenBazaar	openbazaar-go
346	openmicroscopy	openmicroscopy
347	rails	webpacker
348	WikiEducationFoundation	WikiEduDashboard

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
349	docker	compose
350	containers	storage
351	GeoNode	geonode
352	thoughtbot	administrate
353	alphagov	notifications-api
354	mozilla	balrog
355	kiegroup	kie-wb-common
356	grpc	grpc-java
357	matrix-org	synapse
358	dCache	dcache
359	hashicorp	consul
360	artefactual	archivematica
361	TryGhost	Ghost
362	openaddresses	openaddresses
363	bigbluebutton	bigbluebutton
364	frappe	frappe
365	jquery	jquery-ui
366	tensorflow	models
367	biolab	orange3
368	adazzle	react-data-grid
369	IgniteUI	igniteui-angular
370	tablexi	nucore-open
371	commons-app	apps-android-commons
372	digital-asset	daml

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
373	boto	botocore
374	python-pillow	Pillow
375	navikt	foreldrepengesoknad
376	Zeit	next.js
377	manahl	arctic
378	kaltura	playkit-android
379	solidusio	solidus
380	mishoo	UglifyJS2
381	projectatomic	osbs-client
382	bridgedotnet	Bridge
383	Microsoft	BotFramework-WebChat
384	consul	consul
385	querydsl	querydsl
386	apache	cloudstack
387	oaeproject	Hilary
388	restic	restic
389	refinery-platform	refinery-platform
390	OCA	stock-logistics-warehouse
391	linuxmint	Cinnamon
392	metabrainz	picard
393	alphagov	content-performance-manager
394	CONNECT-Solution	CONNECT
395	docker	swarmkit
396	docker	docker-py

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
397	Azure	azure-sdk-for-python
398	Microsoft	AppCenter-SDK-Android
399	marmelab	react-admin
400	learningequality	kolibri
401	SpongePowered	SpongeAPI
402	wildfly	wildfly-core
403	nuxeo	nuxeo
404	LBNL-UCB-STI	beam
405	nextcloud	spreed
406	Skyscanner	backpack
407	typetools	checker-framework
408	networkx	networkx
409	milessabin	shapeless
410	spree	spree
411	cloudify-cosmo	cloudify-manager
412	wix	wix-style-react
413	Blockrazor	blockrazor
414	geotools	geotools
415	home-assistant	home-assistant-polymer
416	openfaas	faas
417	log2timeline	plaso
418	pydanny	cookiecutter-django
419	kuzzleio	kuzzle
420	uber	cadence

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
421	frappe	bench
422	wso2	carbon-identity-framework
423	alphagov	digitalmarketplace-frameworks
424	Wikia	selenium-tests
425	pentaho	pentaho-platform-plugin-common-ui
426	google	cadvisor
427	alv-ch	jobroom2
428	AugurProject	augur-node
429	cloudfoundry-incubator	stratos
430	pentaho	pentaho-platform-plugin-reporting
431	SmarterApp	RDW_Reporting
432	ether	etherpad-lite
433	ajaxorg	ace
434	telstra	open-kilda
435	CartoDB	cartodb
436	withspectrum	spectrum
437	cloudfoundry	cf-deployment
438	palantir	atlasdb
439	18F	identity-idp
440	hibernate	hibernate-ogm
441	aws	aws-cli
442	lerna	lerna
443	DivanteLtd	vue-storefront
444	balderdashy	sails

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
445	go-gitea	gitea
446	prometheus	alertmanager
447	azkaban	azkaban
448	golemfactory	golem
449	hibernate	hibernate-validator
450	Katello	hammer-cli-katello
451	google	gapid
452	foosel	OctoPrint
453	Netflix	Hystrix
454	freeipa	freeipa
455	openlayers	openlayers
456	errbit	errbit
457	pixijs	pixi.js
458	robotframework	SeleniumLibrary
459	alphagov	service-manual-publisher
460	Informasjonsforvaltning	fdk
461	superdesk	superdesk-client-core
462	jeremyevans	sequel
463	mendersoftware	gui
464	CESNET	perun
465	scalacenter	bloop
466	alphagov	pay-selfservice
467	scalaz	scalaz
468	scala-js	scala-js

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
469	LessWrong2	Lesswrong2
470	splicemachine	spliceengine
471	reactjs	reactjs.org
472	common-workflow-language	user_guide
473	FreeCodeCamp	freecodecamp
474	pentaho	pentaho-reporting
475	parse-community	parse-server
476	moment	moment
477	apache	storm
478	pentaho	mondrian
479	Kronos-Integration	kronos-step-archive-tar
480	bonitasoft	bonita-doc
481	pouchdb	pouchdb
482	Gallopsled	pwntools
483	docker-java	docker-java
484	nolanlawson	pinafore
485	ControlSystemStudio	cs-studio
486	naparuba	shinken
487	vmware	vic
488	MovingBlocks	Terasology
489	grpc	grpc-node
490	kaltura	mwEmbed
491	QCoDeS	Qcodes
492	hydroshare	hydroshare

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
493	biocore	scikit-bio
494	nteract	nteract
495	sul-dlss	argo
496	appium	appium-android-driver
497	elastic	apm-server
498	snapcore	snapcraft
499	electron	libchromiumcontent
500	odoomrp	odoomrp-wip
501	chef	omnibus-software
502	purpleidea	mgmt
503	pentaho	pentaho-commons-gwt-modules
504	openpitrix	openpitrix
505	ca-cwds	cals-api
506	osrg	gobgp
507	ledgersmb	LedgerSMB
508	PokemonGoF	PokemonGo-Bot
509	ministryofjustice	prison-visits-2
510	oat-sa	extension-tao-itemqti
511	bazelbuild	bazel
512	nuxt	nuxt.js
513	Alignak-monitoring	alignak
514	yahoo	react-intl
515	mwaskom	seaborn
516	wevote	webapp

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
517	stympy	faker
518	tgriesser	knex
519	avocado-framework-tests	avocado-misc-tests
520	VOLTTRON	volttron
521	typelevel	cats
522	forcedotcom	SalesforceMobileSDK-Android
523	vector-im	riot-web
524	photonstorm	phaser
525	twitter	scalding
526	Caltech-IPAC	firefly
527	gocql	gocql
528	badges	shields
529	containous	traefik
530	decidim	decidim
531	deepchem	deepchem
532	avocado-framework	avocado-vt
533	Kronos-Integration	kronos-adapter-inbound-file
534	openfisca	openfisca-france
535	Adobe-Consulting-Services	acs-aem-commons
536	PrismJS	prism
537	pypa	pipenv
538	goadesign	goa
539	rodjek	puppet-lint
540	wagtail	wagtail

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
541	stamparm	MalTrail
542	vpulim	node-soap
543	CU-CommunityApps	cu-kfs
544	theforeman	foreman_openscap
545	atlasapi	atlas
546	secdev	scapy
547	pyload	pyload
548	INRIA	spoon
549	rstudio	rstudio
550	chanzuckerberg	idseq-web
551	cms-sw	genproductions
552	ExchangeUnion	xud
553	DataDog	integrations-core
554	pentaho	data-access
555	GoogleCloudPlatform	PerfKitBenchmarker
556	hpcc-systems	Visualization
557	marshmallow-code	marshmallow
558	openstreetmap	iD
559	smartcontractkit	chainlink
560	docker	distribution
561	GoogleChrome	workbox
562	Azure	azure-sdk-for-go
563	PulpQE	pulp-smash
564	WorldBank-Transport	DRIVER

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
565	Azure	azure-cli
566	allenai	allennlp
567	alisw	ali-bot
568	wso2	carbon-kernel
569	fishtown-analytics	dbt
570	geoserver	geoserver
571	coredns	coredns
572	Livefyre	streamhub-sdk
573	apache	groovy
574	batfish	batfish
575	theforeman	foreman_discovery
576	apache	zookeeper
577	samaaron	sonic-pi
578	apache	tinkerpop
579	ipfs	go-ipfs
580	codeclimate	codeclimate
581	sanger	sequencescape
582	pubnub	chat-engine
583	fastify	fastify
584	OCA	sale-workflow
585	zenoss	ZenPacks.zenoss.Microsoft.Windows
586	coreos	ignition
587	liferay	liferay-portal
588	airbnb	streamalert

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
589	dcos	dcos-website
590	ultrabug	py3status
591	aodn	aodn-portal
592	Coursemology	coursemology2
593	ManageIQ	manageiq-ui-service
594	jshint	jshint
595	zendesk	maxwell
596	cloudwan	gohan
597	wireapp	wire-android
598	getnikola	nikola
599	mozilla	remo
600	JetBrains	kotlin-web-site
601	python	typeshed
602	miekg	dns
603	mesosphere	marathon
604	biocore	Qiita
605	xenserver	planex
606	vuejs	vuejs.org
607	shotgunsoftware	tk-core
608	Kronos-Integration	kronos-step-stdio
609	sensu	sensu-go
610	publiclab	plots2
611	unlock-protocol	unlock
612	Microsoft	mssql-jdbc

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
613	RocketChat	Rocket.Chat.Android
614	kubevirt	kubevirt
615	prometheus	node_exporter
616	googleapis	nodejs-common-grpc
617	mne-tools	mne-python
618	ruby	ruby
619	slick	slick
620	zendesk	ruby-kafka
621	giantswarm	aws-operator
622	mendersoftware	mender
623	getsentry	raven-ruby
624	ceph	teuthology
625	DSpace	DSpace
626	josdejong	mathjs
627	rspec	rspec-expectations
628	thoughtbot	suspenders
629	mozilla	pontoon
630	gravitational	teleport
631	geoadmin	mf-geoadmin3
632	Dask	dask
633	Yelp	Tron
634	Strapi	strapi
635	guardian	membership-frontend
636	jashkenas	backbone

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
637	mozilla-services	react-jsonschema-form
638	PrairieLearn	PrairieLearn
639	drone	drone
640	HabitRPG	habitica
641	alibaba	pouch
642	JabRef	jabref
643	Hacker0x01	react-datepicker
644	googleapis	google-cloud-java
645	ca-cwds	intake
646	travis-ci	travis-api
647	mattermost	mattermost-redux
648	opensds	opensds
649	eXist-db	exist
650	onosproject	onos-config
651	openfoodfoundation	openfoodnetwork
652	edx	course-discovery
653	raster-foundry	raster-foundry
654	osmandapp	Osmand
655	AzureAD	azure-activedirectory-library-for-android
656	heketi	heketi
657	apache	parquet-mr
658	mapbox	mapbox-java
659	taskcluster	taskcluster-tools
660	wso2	carbon-apimgt

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
661	travis-ci	travis-cookbooks
662	getlantern	lantern
663	steemit	condenser
664	pydata	xarray
665	Azure	aks-engine
666	wix	wix-ui
667	puppetlabs	puppetlabs-postgresql
668	guidance-guarantee-programme	pension_guidance
669	spotify	scio
670	nerdvegas	rez
671	vaadin	flow
672	puppetlabs	bolt
673	dcos	dcos-cli
674	OCA	purchase-workflow
675	18F	cg-dashboard
676	chronologic	eth-alarm-clock-dapp
677	hawkular	hawkular-metrics
678	crowbar	crowbar-core
679	cython	cython
680	mesosphere	dcos-docs-site
681	confluentinc	ksql
682	facebook	buck
683	CruGlobal	give-web
684	QISKit	qiskit-terra

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
685	tidepool-org	blip
686	rmosolgo	graphql-ruby
687	fog	fog-google
688	rollup	rollup
689	chpladmin	chpl-api
690	dwavesystems	dimod
691	strimzi	strimzi-kafka-operator
692	yarnpkg	yarn
693	libopenstorage	openstorage
694	svaarala	duktape
695	puppetlabs	puppetlabs-ntp
696	gohugoio	hugo
697	qutebrowser	qutebrowser
698	apache	activemq-artemis
699	dropwizard	dropwizard
700	NativeScript	nativescript-angular
701	mapstruct	mapstruct
702	zooniverse	Panoptes
703	GPII	universal
704	square	okio
705	Pupil-Labs	Pupil
706	travis-ci	travis-web
707	kubevirt	containerized-data-importer
708	jhipster	jhipster.github.io

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
709	rentpath	react-ui
710	neos	neos-development-collection
711	girder	girder
712	apache	metron
713	liquibase	liquibase
714	grpc	grpc-go
715	strongloop	loopback-datasource-juggler
716	alphagov	digitalmarketplace-supplier-frontend
717	spinnaker	spinnaker
718	bitrise-io	bitrise-steplib
719	codevise	pageflow
720	weaveworks	scope
721	jbosstools	jbosstools-website
722	hashicorp	vault
723	tardis-sn	tardis
724	yahoo	fili
725	ankidroid	Anki-Android
726	kaltura	kmc-ng
727	webdetails	cdf
728	Spesmilo	Electrum
729	mybatis	mybatis-3
730	kivy	python-for-android
731	mozilla	mozregression
732	MiniShift	minishift

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
733	prebid	Prebid.js
734	xapi-project	xs-opam
735	Evennia	evennia
736	OpenRefine	OpenRefine
737	schul-cloud	schulcloud-client
738	busyorg	busy
739	mapbox	rasterio
740	mochajs	mocha
741	invoiceninja	invoiceninja
742	Edraak	edx-platform
743	fluent	fluentd
744	edx	edx-app-android
745	ampproject	amphtml
746	gooddata	gooddata-ruby
747	Octokit	octokit.rb
748	edx	edx-analytics-pipeline
749	googleapis	nodejs-pubsub
750	tronprotocol	java-tron
751	mongodb	docs-bi-connector
752	ray-project	ray
753	pazz	alot
754	thoughtbot	shoulda-matchers
755	weldr	lorax
756	betagouv	mes-aides-ui

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
757	Zeit	hyper
758	translate	pootle
759	kiegroup	jbpm
760	ornicar	lila
761	wso2	product-apim
762	adafruit	Adafruit_Learning_System_Guides
763	OCA	product-attribute
764	Kronos-Integration	kronos-adapter-outbound-file
765	mulesoft	mule-integration-tests
766	openMF	android-client
767	mozilla-iot	gateway
768	taigaio	taiga-back
769	gemini-hlsw	ocs
770	bullhorn	novo-elements
771	kaaproject	kaa
772	plotly	plotly.js
773	Unidata	thredds
774	JeroenDeDauw	Maps
775	binary-com	mobile
776	edx	ecommerce
777	OCA	l10n-switzerland
778	broadinstitute	picard
779	awslabs	sockeye
780	keras-team	keras

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
781	mockito	mockito
782	decentraland	marketplace
783	UniversalLogin	UniversalLoginSDK
784	yahoo	athenz
785	chanzuckerberg	cellxgene
786	googleapis	google-cloud-ruby
787	OHDSI	WebAPI
788	OpenGamma	Strata
789	CenterForOpenScience	ember-osf
790	census-instrumentation	opencensus-go
791	erikras	redux-form
792	TasteJS	todomvc
793	mail-in-a-box	mailinabox
794	coala	coala
795	apache	cordova-lib
796	mytardis	mytardis
797	Yelp	paasta
798	OCA	account-financial-tools
799	OCA	web
800	strongloop	loopback
801	release-engineering	pom-manipulation-ext
802	transloadit	uppy
803	CorfuDB	CorfuDB
804	restify	node-restify

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
805	rpm-software-management	dnf
806	xtuc	webassemblyjs
807	reactjs	react-rails
808	DHIS2	dashboards-app
809	micrometer-metrics	micrometer
810	uktrade	data-hub-frontend
811	Azure	azure-sdk-for-java
812	joblib	joblib
813	DataDog	chef-datadog
814	auth0	cosmos
815	elastic	apm-integration-testing
816	nengo	nengo
817	opensciencegrid	htcondor-ce
818	weecology	retriever
819	quarkusio	quarkus
820	sbt	sbt
821	OCA	website
822	kvhnuke	etherwallet
823	algolia	instantsearch.js
824	meanjs	mean
825	eclipse	jetty.project
826	skycocker	chromebrew
827	embark-framework	embark
828	IQSS	dataverse

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
829	appcelerator	alloy
830	fthomas	refined
831	zalando	skipper
832	rspec	rspec-mocks
833	alphagov	content-tagger
834	kivy	kivy
835	projectatomic	atomic
836	jaegertracing	jaeger
837	datastax	java-driver
838	Kronos-Integration	kronos-cluster-node
839	carrierwaveuploader	carrierwave
840	inorichi	tachiyomi
841	ArduPilot	ardupilot_wiki
842	superdesk	superdesk-core
843	bitzesty	qae
844	mozilla	normandy
845	kontena	pharos-cluster
846	coala	coala-bears
847	Katello	katello-installer
848	matllubos	django-is-core
849	evancohen	smart-mirror
850	Cog-Creators	Red-DiscordBot
851	ConsenSys	kauri-frontend
852	gulpjs	gulp

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
853	freechipsproject	rocket-chip
854	aws	aws-sdk-go
855	rack	rack
856	OCA	l10n-brazil
857	explosion	spaCy
858	dradis	dradis-ce
859	google	syzkaller
860	mozilla-services	socorro
861	edx	devstack
862	nasa	cumulus
863	MDAnalysis	mdanalysis
864	lucas-clemente	quic-go
865	maestrano	mno-enterprise
866	mozilla	MozDef
867	uw-it-aca	myuw
868	beetbox	beets
869	nats-io	gnatsd
870	Azure	WALinuxAgent
871	mozilla	hubs
872	dart-lang	site-webdev
873	buildkite	agent
874	geonetwork	core-geonetwork
875	ReactTraining	react-router
876	exhi	hoist-react

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
877	salesforce	refocus
878	TwilioDevEd	api-snippets
879	status-im	status-go
880	expfactory	expfactory-experiments
881	rubygems	rubygems.org
882	thoughtbot	guides
883	pentaho	pentaho-metadata-editor
884	github	hub
885	OpenZeppelin	openzeppelin-solidity
886	couchbase	sync_gateway
887	guardian	dotcom-rendering
888	wordpress-mobile	WordPress-FluxC-Android
889	puppetlabs	puppetlabs-firewall
890	filecoin-project	go-filecoin
891	mikel	mail
892	prometheus	tsdb
893	openstax	openstax-cms
894	zotonic	zotonic
895	DFE-Digital	manage-courses-backend
896	Microsoft	AdaptiveCards
897	obspy	obspy
898	shakacode	react_on_rails
899	Firebase	FirebaseUI-Android
900	google	closure-library

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
901	hyperledger	sawtooth-next-directory
902	gae-init	gae-init
903	asciidoctor	asciidoctorj
904	puppetlabs	puppetlabs-mysql
905	scrapinghub	portia
906	quantumlib	OpenFermion
907	allegro	hermes
908	Coveo	react-vapor
909	GoogleCloudPlatform	golang-samples
910	the-blue-alliance	the-blue-alliance-android
911	Azure	azure-iot-sdk-node
912	EyeSeeTea	malariapp
913	psychopy	psychopy
914	puppetlabs	puppetlabs-stdlib
915	gammapy	gammapy
916	rubygems	rubygems
917	mulesoft	mule-extensions-api
918	Rapptz	discord.py
919	PrincetonUniversity	PsyNeuLink
920	zulip	python-zulip-api
921	angular	zone.js
922	webpack	webpack-dev-server
923	googleapis	google-cloud-python
924	video-dev	hls.js

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
925	Polymer	tools
926	OCA	l10n-italy
927	serverless	site
928	LiskHQ	lisk-hub
929	moira-alert	moira
930	webdetails	cpf
931	middleman	middleman
932	cdapio	cdap
933	Bcfg2	bcfg2
934	NativeScript	android-runtime
935	hashicorp	packer
936	spring-cloud	spring-cloud-stream
937	indico	indico
938	Jasig	uPortal-start
939	EdgeApp	edge-react-gui
940	internetarchive	openlibrary
941	fabric8-analytics	fabric8-analytics-common
942	soimort	you-get
943	mozilla	foundation.mozilla.org
944	ros-infrastructure	ros_buildfarm
945	sevntu-checkstyle	sevntu.checkstyle
946	nvaccess	nvda
947	tensorflow	cleverhans
948	rom-rb	rom

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
949	w3c	respec
950	graphql-java	graphql-java
951	hmrc	employee-expenses-frontend
952	AmericanMedicalAssociation	ama-style-guide-2
953	datastax	spark-cassandra-connector
954	bgruening	galaxytools
955	edx	XBlock
956	openhab	openhab-docs
957	Shopify	sarama
958	antha-lang	antha
959	kentcc	kentgov
960	projectcypress	cypress
961	sass	node-sass
962	apereo	cas
963	rackerlabs	zoolander
964	jMonkeyEngine	jmonkeyengine
965	godaddy	kubernetes-client
966	gunthercox	ChatterBot
967	googleapis	nodejs-vision
968	sonm-io	core
969	rspec	rspec-rails
970	stephenmcd	mezzanine
971	nginxinc	kubernetes-ingress
972	fusioninventory	fusioninventory-for-glpi

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
973	helm	helm
974	yannickcr	eslint-plugin-react
975	dcos	cosmos
976	activeadmin	activeadmin
977	Microsoft	hcsshim
978	TryGhost	Ghost-Admin
979	google	ExoPlayer
980	pallets	werkzeug
981	pyinstaller	pyinstaller
982	MetaMask	metamask-extension
983	fossasia	open-event-server
984	spf13	cobra
985	ChartIQ	finsemble-seed
986	gwastro	pycbc
987	theforeman	hammer-cli-foreman
988	prisma	prisma
989	octobox	octobox
990	DevExpress	devextreme-reactive
991	drud	ddev
992	fedora-python	portingdb
993	cloudfoundry-incubator	multiapps-controller
994	SAP	fundamental-react
995	gitcoinco	code_fund_ads
996	chef	supermarket

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
997	spring-cloud	spring-cloud-dataflow-ui
998	goatslacker	alt
999	bitcoinjs	bitcoinjs-lib
1000	jay0lee	GAM
1001	hopshadoop	hops
1002	OHDSI	Atlas
1003	pkp	omp
1004	ebmdatalab	openprescribing
1005	SpriteLink	NIPAP
1006	mackerelio	mackerel-agent
1007	algolia	react-instantsearch
1008	scalaz	scalaz-zio
1009	OperationCode	front-end
1010	autolab	Autolab
1011	apache	incubator-openwhisk-wskdeploy
1012	pilosa	pilosa
1013	scalameta	scalameta
1014	jbosstools	jbosstools-server
1015	teamleadercrm	ui
1016	MongoEngine	mongoengine
1017	dnanexus	dx-toolkit
1018	snowplow	snowplow
1019	testing-cabal	testtools
1020	aio-libs	aiohttp

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
1021	apache	cxf
1022	gomods	athens
1023	siacs	Conversations
1024	NVIDIA	DIGITS
1025	Sirupsen	logrus
1026	theforeman	foreman_remote_execution
1027	mongodb	specifications
1028	CiscoDevNet	ydk-gen
1029	Oracle	helidon
1030	dagster-io	dagster
1031	belaban	JGroups
1032	nodegit	nodegit
1033	grafana	metrictank
1034	sigmavirus24	github3.py
1035	jitsi	jitsi-videobridge
1036	heroku	heroku-buildpack-ruby
1037	OCA	pos
1038	intljusticemission	react-big-calendar
1039	spring-cloud	spring-cloud-netflix
1040	Kronos-Integration	kronos-flow
1041	RedHatInsights	insights-frontend-components
1042	sensu	sensu-docs
1043	mulesoft	apikit
1044	OCA	account-financial-reporting

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
1045	jenkinsci	git-client-plugin
1046	betagouv	pass-culture-browser
1047	samsung	GearVRf
1048	mozilla	fxa
1049	cernopendata	opendata.cern.ch
1050	artefactual	archivematica-storage-service
1051	Firebase	firebase-js-sdk
1052	navikt	fpsak-frontend
1053	aporeto-inc	trireme-lib
1054	googleapis	nodejs-bigquery
1055	autotest	autotest
1056	fabric8-launcher	launcher-frontend
1057	haiwen	seadroid
1058	chef	ohai
1059	OCA	partner-contact
1060	googleapis	nodejs-error-reporting
1061	RPTools	maptool
1062	operator-framework	operator-lifecycle-manager
1063	sourcegraph	go-langserver
1064	tornadoweb	tornado
1065	Tiny-Hands	tinyhands
1066	projectcalico	calicoctl
1067	neo4j-contrib	neo4j-graph-algorithms
1068	twitter	finagle

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
1069	greymass	eos-voter
1070	libretro	libretro-database
1071	yast	yast-bootloader
1072	panoptes	POCS
1073	onelogin	ruby-saml
1074	asciidoctor	asciidoctor-pdf
1075	lib	pq
1076	logstash-plugins	logstash-output-elasticsearch
1077	javers	javers
1078	debezium	debezium
1079	CanonicalLtd	subiquity
1080	clay	amphora
1081	dateutil	dateutil
1082	sindresorhus	got
1083	puppetlabs	puppetlabs-concat
1084	plotly	plotly.py
1085	pytest-dev	pytest-django
1086	necolas	react-native-web
1087	elastic	apm-agent-python
1088	SetProtocol	set-protocol-contracts
1089	mesosphere	mesos-dns
1090	airbnb	enzyme
1091	java-native-access	jna
1092	Kronos-Integration	kronos-service-admin

Table A.1 continued from previous page

#	Project Owner Name	Respository Name
1093	fsouza	go-dockerclient
1094	kinecosystem	kinit-android
1095	SUSE	rmt
1096	googleapis	nodejs-storage
1097	Deepomatic	dmake
1098	folio-org	mod-circulation-storage
1099	fourkitchens	emulsify
1100	crowbar	crowbar-ha