Bounties in Open Source Development on GitHub: A Case Study of Bountysource Bounties

Jiayuan Zhou, Shaowei Wang, Cor-Paul Bezemer, Ying Zou, *Member, IEEE,*
and Ahmed E. Hassan, *Member, IEEE*

Abstract—Due to the voluntary nature of open source software, it can be hard to find a developer to work on a particular task. For example, some issue reports may be too cumbersome and unexciting for someone to volunteer to do them, yet these issue reports may be of high priority to the success of a project. To provide an incentive for implementing such issue reports, one can propose a monetary reward, i.e., a bounty, to the developer who completes that particular task. In this paper, we study bounties in open source projects on GitHub to better understand how bounties can be leveraged to evolve such projects in terms of addressing issue reports. We investigated 5,445 bounties for GitHub projects. These bounties were proposed through the Bountysource platform with a total bounty value of $406,425. We find that 1) in general, the timing of proposing bounties and the bounty-usage frequency are the most important factors that impact the likelihood of an issue being addressed. More specifically, issue reports are more likely to be addressed if they are for projects in which bounties are used more frequently and if they are proposed earlier. 2) The bounty value that an issue report has is the most important factor that impacts the issue-addressing likelihood in the projects in which no bounties were used before. Backers in such projects proposed higher bounty values to get issues addressed. 3) There is a risk of wasting money for backers who invest money on long-standing issue reports.

Index Terms—Software evolution, open source software, Bountysource, bounties, GitHub

1 Introduction

Software projects often use issue tracking systems to store and manage issue reports. Developers or users submit issue reports to report bugs or request new features, and wait for these issues to be addressed. However, some issue reports may never be addressed. For example, developers may avoid addressing issues that they consider too low priority, or difficult to implement. To encourage developers to address such issue reports, a *bounty* can be proposed by one or more *backers* for the issue reports.

A bounty is a monetary reward that is often used in the area of software vulnerabilities. Prior studies examined the impact of bounties on vulnerability discovery [4], [7], [24]. Finifter et al. [4] suggested that using bounties as an incentive to motivate developers to find security flaws is more cost-effective than hiring full-time security researchers.

Bounties are now being used to motivate developers to address issue reports, e.g., to fix bugs, to improve performance, or to add new features. $Bountysource$[1] is a platform for proposing bounties for open source projects across multiple platforms (e.g., GitHub) which currently has more than 46,000 registered developers.[2] Although bounties are used in the issue-addressing process, the relationship between bounties and this process is not yet understood. For example, it is unclear whether a bounty improves the likelihood of an issue being addressed (i.e., the issue-addressing likelihood) in projects. By understanding this relationship, we could provide insights on how to better leverage bounties to evolve open source projects, and on how to improve the usability and effectivity of bounty platforms.

In this paper, we study 3,509 issue reports with 5,445 bounties that were proposed on Bountysource from 1,203 GitHub projects, with a total bounty value of $406,425. We first examine the impact of the frequency of bounties (i.e., the bounty-usage frequency) being used in projects and the timing of proposing bounties on the likelihood of an issue being addressed. We found that:

1) Bounty issue reports are more likely to be addressed in projects which are using bounties more frequently. Backers of the projects in which bounties were not used before proposed higher bounty values to get issues addressed.
2) Bounty issue reports for which bounties were proposed earlier are more likely to be addressed. Additionally, there is a risk of wasting money for backers who invest money on long-standing issue reports.

To understand if there are other factors that have an impact on the issue-addressing likelihood, we use logistic regression to study the relationship between 27 factors (including the timing of proposing a bounty and the bounty-usage frequency of a project) along 4 dimensions (i.e.,

- J. Zhou, S. Wang and A. E. Hassan are with the Software Analysis and Intelligence Lab (SAIL), Queen's University, Kingston, Ontario, Canada. E-mail: jzhou,shaowei,ahmed@cs.queensu.ca
- CP Bezemer is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada E-mail: bezemer@ualberta.ca
- Y. Zou is with the Department of Electrical and Computer Engineering, Queen's University, Kingston, Ontario, Canada. E-mail: ying.zou@queensu.ca
- Shaowei Wang is the corresponding author.

[1] https://www.bountysource.com

[2] http://bit.ly/2RkDeCc

project, issue, bounty, and backer) and the issue-addressing likelihood. We found that for bounty issue reports:

1) The timing of proposing bounties and the bounty-usage frequency are the two most important factors that impact the issue-addressing likelihood.
2) The total bounty value that an issue report has is the most important factor that impacts the issue-addressing likelihood in the first-timer projects.

We also performed a manual study on the addressed bounty issue reports in which the bounty remained unclaimed (i.e., the cases in which bounties were ignored by developers). We found that some developers addressed an issue cooperatively, making it difficult to choose a single developer that would be awarded the bounty. In addition, some developers are not driven by money to address issues.

Based on our findings, we have several suggestions for backers and the Bountysource platform. For example, backers should be cautious when proposing small (i.e., $< \$100$) bounties on long-standing issue reports since the risk of losing the bounty exists. Bounty platforms should consider allowing for splittable multi-hunter bounties.

The rest of the paper is organized as follows. Section 2 presents background information about GitHub and Bountysource. Section 3 describes our data collection process. Section 4 introduces our preliminary study. Section 5 and Section 6 study the impact of the bounty on the issue-addressing likelihood in terms of projects' bounty-usage frequency and the timing of proposing bounties. Section 7 investigates more factors that may potentially affect the issue-addressing likelihood. Section 8 studies the closed-unpaid bounty issue reports and discusses the implications of our study. Section 9 discusses the threats to validity of our study. Section 10 introduces related work. Finally, Section 11 concludes our study.

2 Background

In this section, we briefly introduce the issue tracking system on GitHub and the open source bounty platform Bountysource.

2.1 Issue tracking system on GitHub

The issue tracking system (i.e., ITS) on GitHub helps developers to manage the issue reports of their project. Users and developers can report bugs or request new features by posting an issue report on the issue tracking system. There are two statuses of an issue report: "open" and "closed". "Open" indicates that the issue report is still active and is waiting to be addressed. "Closed" indicates that the issue report has been closed. The most common reason for closing an issue report is that the issue has been addressed, but it could also have other reasons (e.g., duplicated issue reports). Users can attach free-text labels to issue reports to indicate the category of an issue report. An issue report contains a title to summarize the issue and a description that describes the issue in detail. Developers can discuss an issue report by leaving comments, which can include code snippets, links or images to improve the description.

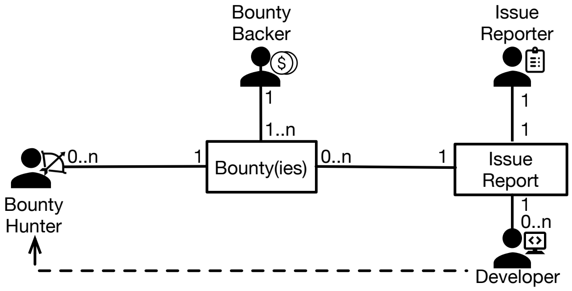


Fig. 1: The relationships between entities that are involved in the bounty process.

2.2 Bountysource

Bountysource is a platform on which users can pledge a monetary incentive (a *bounty*) to address an issue report of an open source project. There exist two roles on Bountysource: the bounty backer and the bounty hunter roles.

Bounty backers, which may be anonymous, are users or developers who propose bounties for issue reports. The backer can set an expiration period for bounties that are over \$100. When the bounty expires, the money is refunded to the backer; otherwise, the bounty stays with the issue report until someone claims it. Bounties that are smaller than \$100 are not refunded if they remain unclaimed. An issue report can have multiple bounties from one or more backers.

Bounty hunters are developers who address issue reports that have bounties. Once a developer claims to have addressed an issue report, its bounty backer(s) can choose to accept (no response will be taken as an acceptance) or reject the claim. In this situation, backers have two weeks to make the decision (accept or reject). If no backer explicitly rejects the claim, the bounties will be paid to the developer automatically. Multiple bounty hunters can work on an issue report at the same time, but the bounties of an issue can only be rewarded to one bounty hunter.

Figure 1 summarizes the relationships between the bounty, the bounty backer, the bounty hunter, the issue report, the issue reporter, and the developer. Basically, when an issue report is submitted by an issue reporter, one or more bounty backers can propose bounty(ies) on the issue report. One or more developers of the issue report can choose to become bounty hunters to address the issue report but only one bounty hunter can get the bounty(ies).

Developers and users from more than 12 platforms (e.g., GitHub) propose bounties for issue reports through Bountysource. In this study, we focus on GitHub issue reports, since the majority of the bounties (see Section 3 for more details) that are proposed on Bountysource are for GitHub issue reports. Figure 2 shows the workflow of the bounty processes between GitHub and Bountysource. The lifecycle of a bounty starts with a bounty backer proposing a bounty for an issue report on GitHub. Bountysource will generate a link to the issue report on GitHub. The bounty backers pledge money to Bountysource (the money is held by Bountysource) and can choose to "advertise" the bounty by tagging the issue report on GitHub with a bounty label (see the example[3] for details), appending the bounty value to the title of the issue report or mentioning the bounty in

[3] http://bit.ly/2EQEA6c

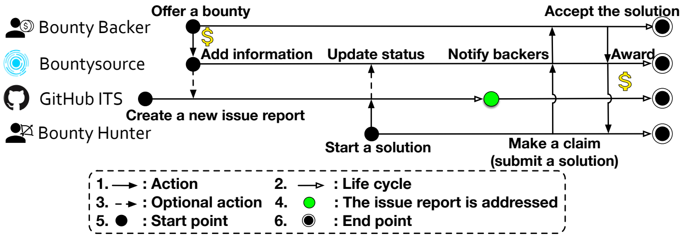


Fig. 2: The workflow of the bounty between GitHub and Bountysource.

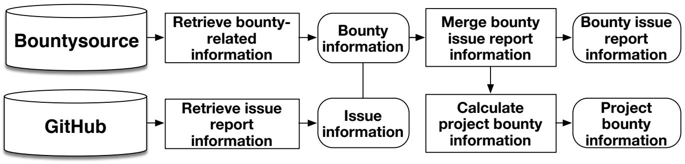

Bountysource
GitHub
Retrieve bounty-related information
Retrieve issue report information
Bounty information
Issue information
Merge bounty issue report information
Calculate project bounty information
Bounty issue report information
Project bounty information

Fig. 3: An overview of our data collection process.

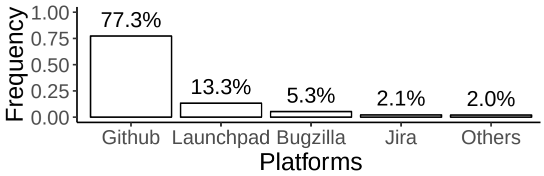


Fig. 4: The distribution of Bountysource bounties across the supported ITSs.

TABLE 1: Dataset description.

Total number of bounties	5,445
Total number of claimed bounties	2,402
Total bounty value	$406,425
Total number of bounty hunters	882
Total number of bounty backers	2,534
Total number of issue reports	3,509
Total number of issue reports with multiple bounties	795
Total number of projects	1,203

the discussion of the issue report in GitHub. When a bounty hunter starts working on an issue, they can update their working status on Bountysource. After the issue report is addressed, the bounty hunter can submit a claim for the bounty on Bountysource and the backer will be notified. Once the bounty backer accepts the claim, the bounty hunter receives the money from Bountysource.

Based on the status of an issue report and whether a bounty is paid out, a bounty issue report has the following three statuses:

Closed-paid: the issue report is closed and the bounty has been successfully rewarded to a bounty hunter. We defined such issue reports as *successful* bounty issue reports.

Open-unpaid: the issue report is open and the bounty is active. We defined such issue reports as *failed* bounty issue reports.

Closed-unpaid: the issue report is closed but the bounty remains unclaimed. We defined such issue reports as *ignored* bounty issue reports.

3 Data collection

In our study, we focus on the bounties that are proposed through the Bountysource platform since it is one of the most popular platforms for open source projects. As explained in Section 2, Bountysource supports issue reports from several ITSs (e.g., GitHub and Bugzilla). Figure 4 shows the distribution of Bountysource bounties across its supported ITSs. The majority of the issue reports come from GitHub (77.3%), hence we focus our study on the bounties that were proposed for GitHub issue reports.

All information about the bounties is stored on Bountysource and all details about issue reports and their corresponding projects are stored on GitHub. Hence, we collected data for our study along three dimensions: the bounty, the issue report, and the project.

Figure 3 presents an overview of our data collection process, which is broken down as follows:

Step 1: We retrieved the bounty and issue information from Bountysource using its official web API.[4] The bounty information includes the backers who proposed the bounty, the proposed bounty value and the hunter who addressed the issue report. In addition, we collected basic information about the GitHub issue reports such as their id and URL.

Step 2: We retrieved the details of the issue reports, which are linked to Bountysource bounties by using the URL and id that we retrieved in step 1, from GitHub using its official web API.[5] For example, we collected the description of the issue report, the creation date of the issue report, the comments that developers left under the issue, and the labels of the issue report.

Step 3: We calculated the corresponding project's bounty information for each collected bounty issue report, such as the number of total bounty issue reports of a project.

In total, we collected 5,445 bounties with a total value of $406,425, together with their corresponding issue reports which were reported between Oct 19, 2012, and Oct 5, 2017. Table 1 gives a description of our dataset.

4 Preliminary Study

In our preliminary study, we first present basic descriptive statistics about the bounty: 1) the distribution of bounty issue reports across the possible statuses (i.e., successful, failed and ignored); 2) the distribution of the number of days between the reporting of the issue and its first bounty being proposed (i.e., *days-before-bounty*); 3) the distribution of the frequency of bounties being used in projects (i.e., bounty-usage frequency). From these statistics, we get a basic view of how bounties are used across projects. In addition, when a bounty issue report is closed and the bounty is paid out, we define this bounty issue report as addressed. We also calculate the issue-addressing likelihood against the bounty value and check the relationship between the issue-addressing likelihood and the bounty value.

62.7% of the bounty issue reports are closed, while the bounties in almost one-third of these closed issue reports remain unpaid with a value of $41,856 in total. Figure 5 shows the distribution of bounty issue reports across the three possible statuses. 37.3% of the bounty issue reports are failed (i.e., open-unpaid). Although 62.7% of the bounty issue reports were closed, almost one-third of their bounties

[4] https://bountysource.github.io/

[5] https://developer.github.com/v3/

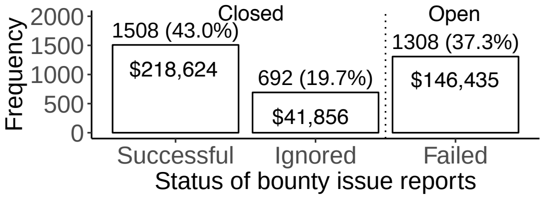


Fig. 5: The distribution of the possible statuses of bounty issue reports and their corresponding cumulative bounty value.

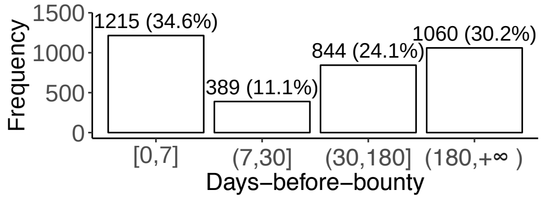


Fig. 6: The distribution of the number of days between the reporting of the issue and its first bounty being proposed (i.e., *days-before-bounty*) in four different time ranges.

were ignored (i.e., closed-unpaid). The total value of the ignored bounties ($41,856) is "frozen" in the Bountysource platform unless someone claims the bounty. In the rest of the paper, when we say the likelihood of an issue being addressed (i.e., issue-addressing likelihood), we only refer to the bounty issue reports that are successful (i.e., closed and paid out). We do not take the issue reports which were ignored into consideration because the hunters might not be driven by the bounty in such issue reports. We conducted a qualitative study of these closed-unpaid bounty issue reports to better understand them in Section 8.1.

34.6% of the bounties were proposed within 7 days from the creation of an issue report, while 30.2% of bounties were proposed after more than 180 days. Figure 6 shows the distribution of the *days-before-bounty* metric in four continuous time ranges (i.e., 0 to 7 days, 8 days to 30 days, 31 days to 180 days, and more than 180 days). We observe that in 34.6% of the issue reports their first bounty was proposed within 7 days after their creation, while in another 30.2% of the issue reports their first bounty was proposed after more than 180 days since their creation.

More than half of the projects only used a bounty once, while two projects used bounties very frequently (more than 100 times). The distribution of the bounty-usage frequency of each project is skewed (with a variance of 57.02). 768 (65%) projects used a bounty only once, 62 projects used a bounty at least 10 times and only 9 projects used a bounty more than 50 times.

The correlation between the bounty value and the issue-addressing likelihood is weak. Figure 7 presents the issue-addressing likelihood of an issue report against the bounty value of an issue report. We do not observe patterns between them. The correlation between the bounty value and the issue-addressing likelihood is surprisingly weak (0.14).

Counter-intuitively, the correlation between the bounty value and the issue-addressing likelihood is weak. However, this low correlation may be due to the variation of the frequency of bounties that were used before in different projects and the timing of proposing the bounty.

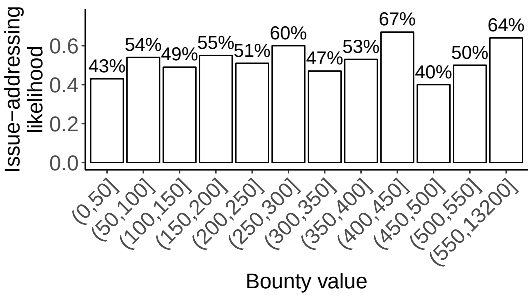


Fig. 7: The issue-addressing likelihood of the proposed bounty value ranges.

Therefore, in Section 5 we investigate how the issue-addressing likelihood changes across projects with different bounty-usage frequencies. We study the relationship between the timing of proposing bounties and the issue-addressing likelihood in Section 6.

5 Studying how the issue-addressing likelihood changes across projects which have different bounty-usage frequencies

Bounties are frequently used in some projects while rarely used in others (see Section 4). The frequency of using bounties in a project may reflect the degree of experience that the project has with bounties and may have an impact on the issue-addressing process. Therefore, in this RQ, we investigate how the issue-addressing likelihood changes across different projects with different bounty-usage frequencies. Furthermore, we study the relationship of the issue-addressing likelihood with the bounty value and the project activities (e.g., creating pull requests). With a better understanding of the bounty across different projects, we can provide insights on how to better use bounties.

Approach: Figure 8 gives an overview of our approach. We elaborate on each step below.

Project categorization: Given the variance (see Section 4) of the bounty-usage frequency for different projects, it is not advisable to study all the issue reports as one group when we study the bounties at the issue level. Therefore we categorize the projects into the following three groups:

1) **First-timer bounty-projects:** Projects which have only one bounty issue report.
2) **Moderate bounty-projects:** Projects which have 2 to 50 bounty issue reports.
3) **Frequent bounty-projects:** Projects which have more than 50 bounty issue reports.

It is important to study the bounties in the first-timer bounty-projects, since users of such projects may not have former bounty experience. Insights on the usage (e.g., how large of a bounty should be proposed?) and the impact (e.g., the issue-addressing likelihood) of bounties in these projects may benefit other projects which are new to using bounties. We grouped the projects that have more than 50 bounty issue reports as well since we assume that in such projects the community is more familiar with the use of bounties. After grouping the projects into the above mentioned three

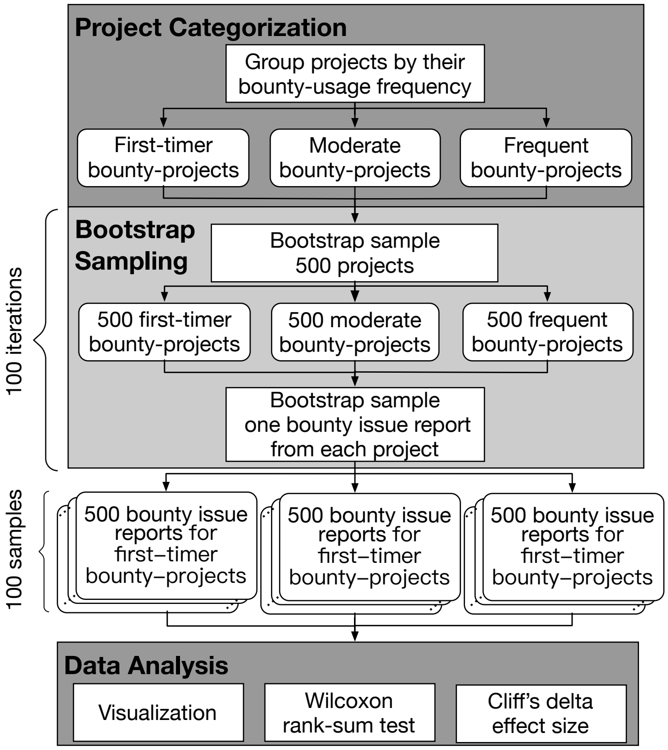

Project Categorization
Group projects by their bounty-usage frequency
First-timer bounty-projects
Moderate bounty-projects
Frequent bounty-projects
Bootstrap Sampling
100 iterations
Bootstrap sample 500 projects
500 first-timer bounty-projects
500 moderate bounty-projects
500 frequent bounty-projects
Bootstrap sample one bounty issue report from each project
100 samples
500 bounty issue reports for first–timer bounty–projects
500 bounty issue reports for first–timer bounty–projects
500 bounty issue reports for first–timer bounty–projects
Data Analysis
Visualization
Wilcoxon rank-sum test
Cliff's delta effect size

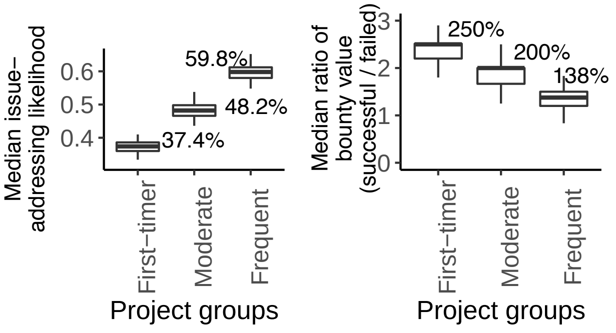
Fig. 8: An overview of our approach.

groups we ended up with 768 (65%) first-timer bounty-projects, 400 (34%) moderate bounty-projects, and 9 (1%) frequent bounty-projects.

Bootstrap sampling: After grouping the projects into the three groups, we used a bootstrap sampling approach to sample issue reports across projects. We used bootstrap sampling to reduce the bias that is caused by the unbalanced number of projects across the three groups. We first randomly sampled 500 projects from each group with replacement. Then we randomly sampled one bounty issue report from each sampled project, to avoid a bias towards projects with more issue reports than other projects in the same group. Hence, we sampled 500 bounty issue reports from each of the 3 project groups. To make our results more reliable, we repeated the sampling process 100 times with different random seeds. We ended up with 100 samples with 1,500 issue reports each (500 issue reports for each group).

Data Analysis: For each sample, we calculated the issue-addressing likelihood across the three project groups and plotted the results. To compare the differences between two data distributions (e.g., the differences of the bounty values between successful and failed bounty issue reports), we used the Wilcoxon rank-sum test [1], which does not require the sample to be normally distributed. We consider the two distributions as significantly different when the p-value of the test is smaller than 0.05. Furthermore, we applied Cliff's delta d [12] effect size to quantify the magnitude of the difference. We use the following thresholds for d [17]: $|d| \leq 0.147$ (negligible); $0.147 < |d| \leq 0.33$ (small); $0.33 < |d| \leq 0.474$ (medium); $0.474 < |d| \leq 1$ (large).

Results: Bounty issue reports in projects which have a higher bounty-usage frequency are more likely to be addressed. Figure 9a shows the median issue-addressing likelihood for each project group. We observed a positive relationship between this likelihood and the bounty-usage frequency, which indicates that a bounty issue report is more likely to be addressed in a project with a higher bounty-usage frequency. Our statistical test shows that the issue-addressing likelihood is significantly different across these three groups of projects with a large effect size. One possible explanation is that the lower bounty-usage frequency of a project may indicate that the backers have less experience in proposing a bounty (e.g., at the proper time with a proper value) and the hunters react to bounties less actively than in projects with a higher bounty-usage frequency.



(a) (b)

Fig. 9: The distribution of the median issue-addressing likelihood (a) and the distribution of the median ratio of the bounty value of successful bounty issue reports to the value of failed bounty issue reports (b) for each project group for 100 samples.

The successful bounty issue reports have a relatively higher bounty value than failed bounty issue reports, particularly in the first-timer bounty-projects. Figure 9b shows that the median ratio (the bounty values of successful reports compared to those of failed reports) is larger than 1 in all 3 project groups. In other words, the successful bounty issue reports have higher bounty values than the failed bounty issue reports. In addition, comparing the ratio across project groups, the first-timer bounty-projects have the largest ratio (2.5) among the three groups, indicating that developers may want more money to address an issue in first-timer bounty-projects than in other projects. Our statistical test results shows that the differences in bounty value between successful and failed bounty issue reports are significant ($p\text{-value} < 0.05$) in the first-timer bounty-projects for all samples with a non-negligible Cliff's delta effect size (72% samples have a small effect size and 28% samples have a medium effect size). In 97% of the samples of the moderate bounty-projects, the bounty value of successful issue reports is significantly higher than that of failed issues (with a non-negligible effect size in 77% of the cases). For the frequent bounty-projects, in 92% of the samples, the differences were not statistically significant.

One possible explanation is that first-timer bounty-projects may not be as active as moderate and frequent projects. Therefore, backers would be required to propose bounties with higher values to attract enough attention from the community for addressing issues. To investigate this explanation, we examined the activity of the projects in terms of the number of pull requests, issue reports, and commits. Figure 10 shows the distributions of the occurrences of these three activities in each project group. **Projects with fewer**

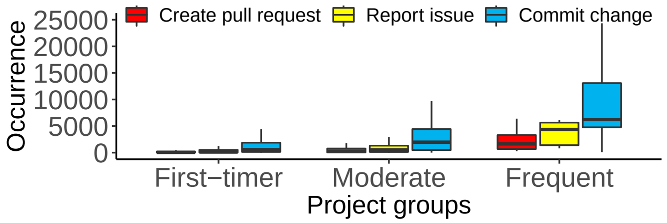


Fig. 10: The distributions of the occurrences of three activities (i.e., the create pull request, the report issue and the commit change) in each project group.

bounty issue reports are usually less active (in terms of the number of pull requests, issue reports, and commits) than projects with more bounty issue reports. Another possible explanation is that the backers in the first-timer bounty-projects have no experience in proposing bounties and sometimes overestimate the value of addressing an issue report. In this situation, the overestimated bounty issue reports may attract more attention from the community due to their "easy money" and get addressed easier.

> *The issue-addressing likelihood is higher in projects which used bounties more frequently. Successful bounty issue reports have higher bounty values than failed bounty issue reports in the projects which used bounties less frequently.*

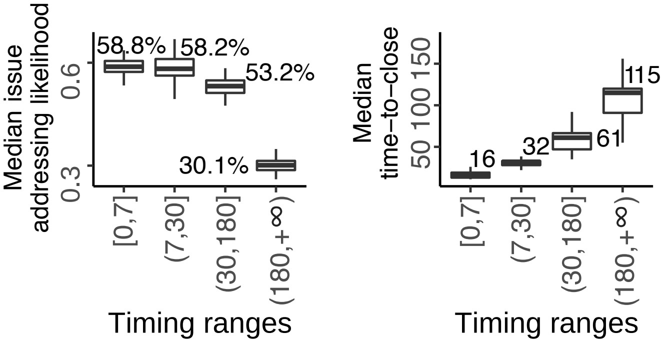
6 Studying the Relationship Between the Timing of Proposing a Bounty and the Issue-Addressing Likelihood

In Section 4, we observe different patterns for the timing of proposing bounties, which may have different relationships with the issue-addressing likelihood. In this section, we investigate the relationship between the timing of proposing bounties and the issue-addressing likelihood. With a better understanding of the relationship, we can provide insights into how to improve the timing of proposing a bounty.

Approach: We use the same data (100 samples of 1,500 issue reports for each sample) as RQ1. For each sample, we grouped data into four timing ranges based on the number of elapsed days between the creation of an issue report and the proposal of its first bounty(i.e., *days-before-bounty*). We defined the four timing ranges as follows:

1) **[0,7]:** the first bounty was proposed within seven days after the issue was reported.
2) **(7,30]:** the first bounty was proposed between 7 and 30 days after the issue was reported.
3) **(30,180]:** the first bounty was proposed between 30 and 180 days after the issue was reported.
4) **(180,∞):** the first bounty was proposed after 180 days after the issue was reported.

Then we calculate the issue-addressing likelihood across the timing ranges for each sample to study the relationship between the timing and issue-addressing likelihood. To further study the relationship between the issue-addressing speed and the timing, we also calculate the time it takes to close the issue report after the first bounty is proposed (i.e., *time-to-close*) in each timing range for each sample. Furthermore, we also study the timing in terms of different project groups for each sample.



(a) (b)

Fig. 11: The distribution of the median issue-addressing likelihood (a) and the distribution of the median time to close an issue report (i.e., *time-to-close*) (b) across the timing ranges for the 100 studied samples.

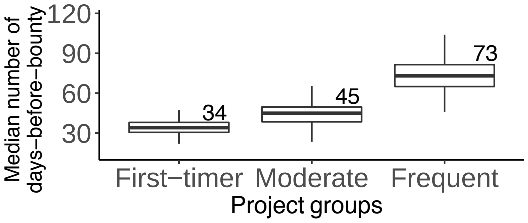


Fig. 12: The distribution of the median number of days between the creation of the issue report and the proposed of the first bounty to the bounty is proposed (i.e., *days-before-bounty*) across different project groups for 100 samples.

Results: In general, issue reports for which bounties were proposed earlier have a higher likelihood of being addressed. Figure 11a shows the distribution of the median issue-addressing likelihood across the timing ranges over 100 samples. We calculated the Wilcoxon rank-sum test and Cliff's delta test to measure the differences between two distributions. The results show that there is no significant difference between the distributions of "[0,7]" and "(7,30]" while there exist significant differences with large effect sizes for the other two distributions. The likelihood is getting smaller as the *days-before-bounty* gets larger, especially for the issue reports in which bounties are proposed after 180 days where the likelihood drops to 30%.

One possible explanation is that as time progresses, the risk of a report becoming obsolete exists, leaving the issue report unaddressed even after a bounty is proposed. For example, an issue report[6] that was created on Feb 4, 2016 in the `uappexplorer` project requested a new feature for an Ubuntu Phone Application. The owner of the application and another developer both showed great interest in this issue. Because of the lack of time, the feature was never added. A bounty of $5 was proposed[7] after almost one year (i.e., on Jan 12, 2017). However, the issue report was closed because Ubuntu Phone was no longer used making the issue report obsolete. Because of the low value, the bounty was not refundable. In other words, **backers carry the risk of**

[6] https://github.com/bhdouglass/uappexplorer/issues/69
[7] http://bit.ly/2Q3BIns

wasting their money by proposing small bounties on such long-standing issue reports.

Another possible reason for the lower issue-addressing likelihood of the issue reports for which bounties were proposed later is the potential difficulty of such issue reports. Figure 11b shows the median time that was taken to close issues (i.e., *time-to-close*) across each timing range. We observe that the issue reports in which bounties were proposed later took more time to be addressed.

Backers proposed bounties earlier in the first-timer bounty-projects. Figure 12 presents the distribution of the median value of the *days-before-bounty* metric across the project groups for each sample. The median *days-before-bounty* is 34, 45 and 73 for each project group, respectively. We calculated the Wilcoxon rank-sum test and Cliff's delta d effect size to measure the differences of distributions between the first-timer bounty-projects and the moderate bounty-projects, the moderate bounty-projects and the frequent bounty-projects, respectively. The result shows that any two pairs of distributions are significantly different with large Cliff's delta effect sizes, indicating that the *days-before-bounty* is higher in projects with a higher bounty-usage frequency. One explanation is that the activity of first-timer bounty-project is lower (see Figure 10), which may encourage backers to propose a bounty earlier to attract developers to get an issue addressed.

> *In general, issue reports for which bounties were proposed earlier have a higher likelihood of being addressed. In addition, the risk of losing money exists for backers who propose small bounties for long-standing issue reports.*

7 Using a logistic regression model to study the relationship between the studied factors and the issue-addressing likelihood

In the previous sections, we studied the relationship between the issue-addressing likelihood, the bounty-usage frequency and the timing of proposing a bounty. However, there may exist more factors that potentially affect the issue-addressing likelihood. In this section, we conduct a deeper study on the relationship between other factors and the issue-addressing likelihood. With a better understanding of this relationship, we can provide insights into how to use a bounty to improve the issue-addressing process.

7.1 Approach

In this section, we use the same data as in Section 5 to build logistic regression models to investigate the relationship between the studied factors and the issue-addressing likelihood. Firstly, we build models on the entire set of bounty issue reports to understand the global relationship (referred to as the **global model**). Secondly, as we can see in Section 5, the issue-addressing likelihood changes across project groups (i.e., the first-timer bounty-projects, the moderate bounty-projects, and the frequent bounty-projects). To understand the relationship within each project group, we build logistic regression models on the bounty issue reports of each group separately. To condense our writing, we refer to the model for the first-timer, moderate, and frequent bounty-projects as the **first-timer model**, **moderate model**, and **frequent model**, respectively. We use the logistic regression modeling technique since it is a robust and highly interpretable technique, which has been applied successfully in software engineering before [14], [20]. Figure 13 gives an overview of our approach. Below, we elaborate on the studied factors, the processes of the model constructions, and the analysis of our models.

7.1.1 Studied factors

We consider 27 factors along 4 dimensions:

1) **Issue report basic:** Eight factors which can estimate the length and the popularity of an issue report.
2) **Issue report bounty:** Five factors which describe the bounty usage within a bounty issue report.
3) **Project bounty:** Six factors which reflect the bounty usage within a project.
4) **Backer experience:** Eight factors which capture the bounty experience of the backers of a bounty issue report.

Table 2 and Table 3 summarize the description of and rationale behind the studied factors. The factors which are marked with '*' are time-dependent factors which are calculated at the time when the bounty is proposed. Note that the factors in the project bounty, issue report basic, and backer experience dimensions cannot be changed by a backer who wants to propose a bounty. Hence we include these factors as control factors in our models.

7.1.2 Model construction

Similar to prior studies [9], [14], [16], [20], we first removed correlated and redundant factors by using the Spearman rank correlation test and the redundancy analysis to avoid multicollinearity. We ended up with three factors in the project bounty dimension, six factors in the issue report basic dimension, four factors in the issue report bounty dimension, and three factors in the backer experience dimension. Then we added non-linear terms in the model to capture the more complex relationship in the data by employing restricted cubic splines [6]. Finally, we used the `rms R` package[8] to implement our logistic regression models (i.e., the first-timer, moderate, frequent, and global models) based on 100 samples and ended up with 400 models. See our appendix [26] for more details about our model construction.

7.1.3 Model analysis

For each logistic regression model, we used the Area Under the Receiver Operating Characteristic Curve (i.e., *AUC*) and a bootstrap-derived approach [3] to assess the explanatory power of the models following prior studies [9], [14], [20]. The AUC ranges from 0 to 1, with 0.5 being the performance of a random guessing model and a higher AUC meaning that the model has a higher ability to capture the relationships between the explanatory factors and the response factor. To check whether the models are not overfitted, we calculate their *optimism* values using a bootstrap-derived

[8] https://cran.r-project.org/web/packages/rms/index.html

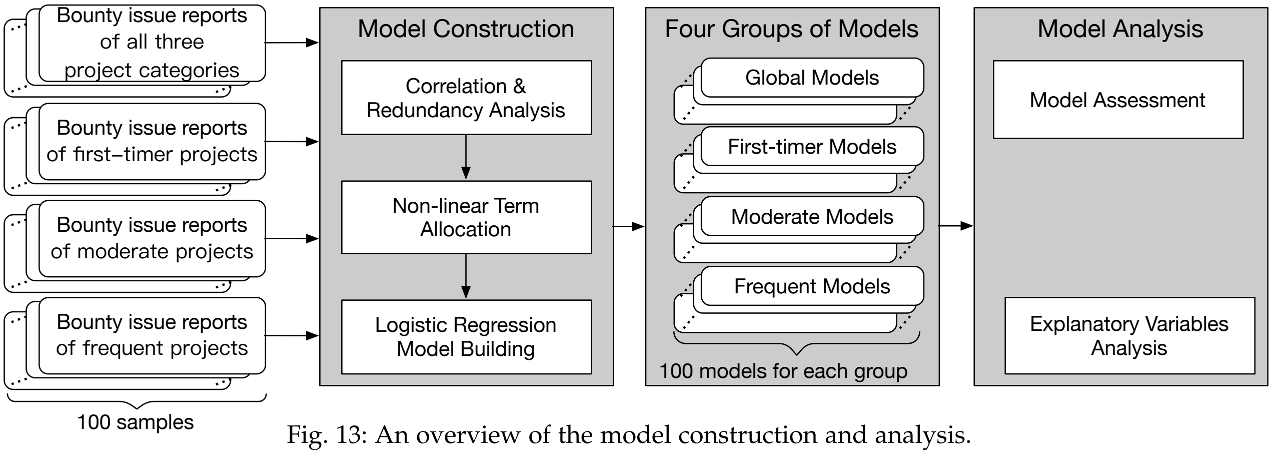

Bounty issue reports of all three project categories
Bounty issue reports of first−timer projects
Bounty issue reports of moderate projects
Bounty issue reports of frequent projects
100 samples
Model Construction
Correlation & Redundancy Analysis
Non-linear Term Allocation
Logistic Regression Model Building
Four Groups of Models
Global Models
First-timer Models
Moderate Models
Frequent Models
100 models for each group
Model Analysis
Model Assessment
Explanatory Variables Analysis

Fig. 13: An overview of the model construction and analysis.

TABLE 2: The description of and rationale for the factors in the *Issue report basic* and the *Issue report bounty* dimensions. The factors which are marked with '*' are time-dependent factors which are calculated at the time when the bounty is proposed

Factor name	**Description**	**Rationale**
Issue report basic		
I_content_len*	The length of an issue report and its comments (in characters).	These factors reflect the amount of supportive information that an issue report has. Issue reports with more supportive information may help developers to address them.
I_code_len*	The total length of the code snippets in an issue report and its comments (in characters).	
I_code_proportion*	The proportion of code in an issue report and comments (i.e., $\frac{I\_code\_len}{I\_content\_len}$).	
I_link_cnt*	The number of links in an issue report and its comments.	The discussion activities reflect the popularity of an issue report, which may have a relationship with the issue-addressing likelihood.
I_img_cnt*	The number of images in an issue report and its comments.	
I_cmnt_cnt	The number of comments that an issue report received.	
I_participant_cnt*	The number of participants in the discussion of an issue.	
I_cmnt_per_day_mean*	The mean number of comments per day for an issue report.	
Issue report bounty	**Description (d) - Rationale (r)**	
I_B_days_before_bounty*	d: The number of days between the creation of an issue report and its first bounty. r: Different timing of proposing bounties have relationship with the issue-addressing likelihood.	
I_B_total_value	d: The total bounty value of the issue report. r: A higher bounty may attract more developers.	
I_B_cnt	d: The number of bounties that a bounty issue report has. r: A higher number indicates that more backers are interested in getting this issue addressed.	
I_B_has_label	d: Whether a bounty issue report is tagged with a bounty label. r: A bounty label could help draw attention from the community, which may impact the issue-addressing likelihood.	
I_B_timing_range*	d: The range of the timing of proposing the first bounty. r: The timing of proposing a bounty has a relationship with the issue-addressing likelihood (see Section 6).	

approach (with a small optimism value indicating that the model is not overfitted).

To study the impact of each factor on the issue-addressing likelihood, we used the **anova** function in the R **rms** package to compute the Wald χ^2 value. The larger the Wald χ^2 value of a factor is, the larger the impact of the factor on the issue-addressing likelihood. We computed the Wald χ^2 value for each factor for each model and used the median Wald χ^2 value of each factor within a group to represent the impact of that factor in that group.

In addition, to further understand how a factor influences the value of the response variables, we used the **Predict** function in the **rms** R package to plot the estimated issue-addressing likelihood against a factor. Since all models across 100 samples showed similar patterns of influence for the factors, we randomly selected a sample as an example to build models and visualize the results (see Figure 14 and Figure 15). The analysis allows us to further understand how a factor affects the issue-addressing likelihood. For the detailed results of our model analysis, such as the χ^2 values, we refer the reader to our online appendix [26].

TABLE 4: The two most important factors, the median AUC and the median optimism (Opt.) values for our models.

Group	**Factor importance**		**Median**	
	1^{st}	2^{nd}	AUC	Opt.
Global	*I_B_timing_range*	*P_B_usage_group*	0.72	0.02
First-timer	*I_B_total_value*	*I_B_timing_range*	0.74	0.03
Moderate	*I_B_timing_range*	*I_code_proportion*	0.71	0.04
Frequent	*I_B_timing_range*	*P_B_paid_proportion*	0.81	0.03

7.2 Results

Our models explain our dataset well and have a reliable performance. The median AUCs for each group of models are all at least 0.71 (see Table 4), which indicates that our models explain the dataset well and the low median optimism values (between 0.02 and 0.04) indicate that our models do not overfit the dataset.

In the global view, the timing of proposing the bounties and the bounty-usage frequency are the top two

TABLE 3: The description of and rationale for the factors in the *Project bounty* and the *Backer experience* dimensions. The factors which are marked with '*' are time-dependent factors which are calculated at the time when the bounty is proposed

Factor name	**Description**	**Rationale**
Project bounty		
P_B_I_cnt* P_B_paid_cnt* P_B_open_cnt* P_B_paid_proportion* P_B_total_value*	The total number of issue reports with at least one bounty of a project. The total number of paid bounty issue reports of a project. The number of open bounty issue reports of a project. The proportion of paid bounty issue reports of a project. The total value of the bounties of a project.	These five factors reflect the bounty activity of the project. A different level of activity may have a different impact on the issue-addressing likelihood in the project.
P_B_usage_group	The group of projects.	Different groups of projects have different issue-addressing likelihoods (see Section 5).
Backer experience		
Backer_exp_B_median/sum/max_value* Backer_exp_B_median/sum/max_cnt*	The median/sum/max value of bounties which the backers of this bounty have ever proposed in the past. The median/sum/max number of bounties which the backers of this bounty have ever proposed in the past.	Bounties from a backer who has proposed bounties often, or proposed high-value bounties in the past may attract more attention from developers. t
Backer_role_any_insider* Backer_role_have_reporter*	Whether any of the backers who has ever contributed to the project. Whether the issue reporter is one of the backers for that issue report.	A backer who has ever interacted with the project before may help the bounty attract more attention from the community.

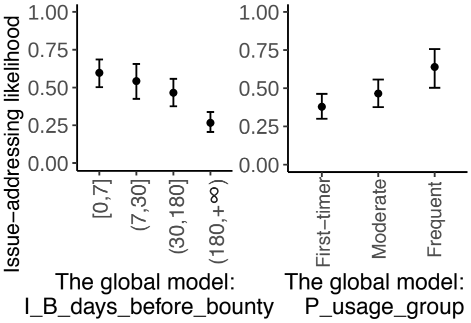


Fig. 14: The plots show the relationship between the studied factors and the issue-addressing likelihood in the global models in the selected sample. For each plot, we adjusted all factors except the studied factor to their median value in the model and recomputed the issue-addressing likelihood. The grey area represents the 95% confidence interval.

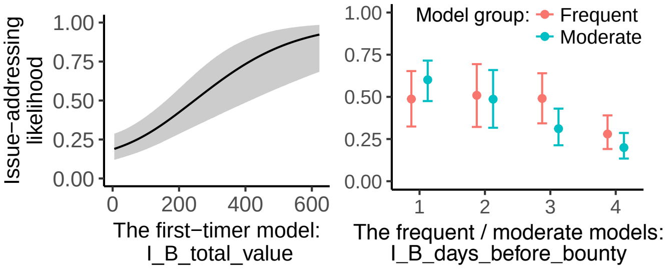


Fig. 15: The plots show the relationship between the studied factors and the issue-addressing likelihood in the first-timer, moderate and frequent models in the selected sample. For each plot, we adjusted all factors except the studied factor to their median value in the model and recomputed the issue-addressing likelihood. The grey area represents the 95% confidence interval.

most important factors that impact the issue-addressing likelihood. Table 4 shows the top two important factors (ranked by median Wald's χ^2 value) in the global models across 100 samples. *I_B_timing_range* (i.e., the range of the timing of proposing bounties) and *P_B_usage_group* (i.e., the group of projects' bounty-usage frequency) are the most important factors which contribute the most explanatory power to the models. This observation echoes our findings in Section 5 and 6 that the timing of proposing bounties and the bounty-usage frequency have a strong impact on the issue-addressing likelihood.

Figure 14 shows the relationship between the issue-addressing likelihood and the top two most important factors for the global model. *I_B_timing_range* has a negative relation with the issue-addressing likelihood, which indicates that issue reports for which bounties are proposed earlier have a higher likelihood of being addressed. This observation echoes with our findings in Section 6.

In the projects that use bounties moderately and frequently, issue reports are more likely to be addressed if backers propose bounties on an issue report earlier. Table 4 shows that *I_B_timing_range* is the most important factor in the moderate and the frequent models. Especially in the moderate models, *I_B_timing_range* contributed more explanatory power than the other factors. For *P_B_usage_group*, we observe a positive relation with the issue-addressing likelihood in Figure 15. One possible explanation is that projects with a higher bounty-usage frequency have more experience in using bounties which may increase the likelihood of a bounty being addressed. For example, the `eslint` project maintains a document on how bounties work[9]. The `eslint` project has 43 successful (i.e., closed-paid) and only one failed (i.e., open-unpaid) bounty issue report.

The total bounty value of an issue report is the most important factor that impacts the issue-addressing likelihood in the first-timer bounty-projects, while it is less important in the projects in which bounties are used more frequently. From Table 4, we can see that *I_B_total_value* (i.e., the total bounty value of a bounty issue report) is the most important factor in the first-time model, while it is not as important in projects in which bounties are

[9] http://bit.ly/2Sql57n

used more frequently. Figure 15 shows a positive relation between *I_B_total_value* and the issue-addressing likelihood of the first-time projects. This observation is compatible with our observations in Section 5 that successful bounty issue reports usually have much higher bounty values than failed bounty issue reports in the first-timer bounty-projects.

In general, the timing of proposing bounties and the bounty-usage frequency are the top two most important factors that impact the issue-addressing likelihood. The total bounty value that an issue report has is the most important factor that impacts the issue-addressing likelihood in the first-timer bounty-projects, while it is not as important for projects in which bounties are frequently used.

8 Discussion

8.1 Studying ignored bounty issue reports

In Section 4, we observed that in 19.7% of the bounty issue reports the bounties were ignored (i.e., closed-unpaid). For these ignored bounty issue reports, the issue reports were closed but the bounties remained unclaimed. It seems that money was not the driver that motivated developers to address these issues. To understand the possible reasons behind this phenomenon, we manually studied all 692 ignored bounty issue reports (with a total bounty value of $41,856). Because the "closed" status of an issue report does not necessarily mean that the issue was addressed (e.g., a report may have been a duplicate of another issue report), it is very difficult to automatically identify whether an issue in the closed issue report was addressed or not. Therefore, we need to manually examine the closed-unpaid bounty issues reports to filter out the reports that were closed for another reason than the issue being addressed.

21.8% (479 out of 2,200) of the addressed bounty issue reports were not paid out. We identified that 479 out of the studied 692 bounty issue reports were closed because the issues were addressed. Such cases are interesting since the developers could have claimed the bounty but they did not. We manually examined the discussion for these 479 issue reports. We identified 19 cases in which developers gave an explanation for not claiming the bounty. We grouped the explanations as follows:

The developer is not driven by money. In 7 out of 19 cases a developer refused to claim the bounty because they were not motivated by money to address the issue. For example, one developer was against the bounty because they felt that the issue-addressing process should be driven by the interests of the community rather than money. A contributor of the `Brython` project,refused the bounty because he wanted to keep `Brython` free from monetary motivations: "*What is this 'bounty' thing? Needless to say, I refuse that anybody (me included, of course) gets paid for anything related to* `Brython`*.*"[10] In addition, he also asked bounty backers to remove all bounties within the `Brython` project although he respected prior bounties that were paid out. There were five bounty issue reports in the `Brython` project and four bounty issue reports that were addressed without claiming the bounty.

The developer is afraid of sending the wrong message. Krishnamurthy et al. [11] pointed out that financial incentives may cause confusion in the community because the financial incentives may drive a project's own product development cycle away from what is in place. We observed that developers expressed similar concerns. A developer of the `Facebook/HHVM` project,explained that: "*That's very generous of you, but I can't accept a bounty for doing my job. :-P It would be a conflict of interest, and I worry it sends the wrong message about how we prioritize issues from the community.*"[11]

The issue report was addressed by more than one developer. We found nine cases where bounties ended up unclaimed because an issue report was addressed by multiple developers cooperatively and they felt inappropriate to claim the bounty by one developer. For example, the issue[12] was addressed by two developers and because a bounty cannot be split into two parts, no one claimed it.

8.2 The implications of our findings

Backers should consider proposing a bounty as early as possible and be cautious when proposing small bounties on long-standing issue reports. The timing of proposing a bounty is an important factor that impacts the issue-addressing likelihood. In Sections 6 and 7, we revealed the fact that issue reports for which bounties were proposed earlier are more likely to be addressed. Additionally, we observed that issue reports for which bounties were proposed earlier are more likely to be addressed faster. Backers benefit from the higher issue-addressing likelihood and faster issue-addressing speed by proposing bounties earlier.

In Section 6, we also noticed a big drop (i.e., from 53.2% to 30.1%) of the issue-addressing likelihood when backers proposed bounties on long-standing (i.e., more than half a year) issue reports. This drop might be due to such issue reports having become obsolete or being hard to address. Since bounties with a value of less than $100 will not be refunded to the backers if the issue report remains unaddressed, we suggest that backers be cautious when proposing small bounties on long-standing issue reports.

Backers should consider proposing a relatively bigger bounty in first-timer bounty-projects. Although the issue-addressing likelihood is only 37.4% for projects with no bounty-usage experience, the first-timer model in Section 7 shows that the bounty value of an issue report is the most important factor in the first-timer projects, as the issue-addressing likelihood is higher for higher bounty values. The high ratio (2.5) of the bounty value of successful bounty issue reports to the bounty value of failed bounty issue reports also supports this finding. We suggest that backers of projects with no bounty-usage experience propose higher bounty values for issue reports.

Bounty platforms should allow for splittable multi-hunter bounties. In addition to a voluntary nature, open source projects have a collaborative nature. Some issues are hard for a developer to address alone. Hence, we encourage developers to work together, especially for issue reports which have a high bounty value (as these issue reports are often harder to address). However, the current

[10] http://bit.ly/2OTYx0x

[11] http://bit.ly/2OZw1uw

[12] http://bit.ly/2PrMiHV

bounty workflow only allows **one** bounty hunter to claim the bounty, which goes against the collaborative nature of open source. It may also drive the developers, who want to collaboratively address the issue, away because not every participant will get a reward at the end. Therefore, bounty platforms should consider adding the ability for a bounty to be split across multiple hunters to encourage developers to work together on difficult bounty issues.

Bounties should be transferable. The total value of all addressed-unpaid bounties ($43,256) is “frozen” in Bountysource. In addition, the median number of days between the closing date of the issue report and the date of collecting our data is 372.5, which means that more than half of the bounties from the ignored bounty issue reports were unclaimed for at least one year. By manually examining these 479 addressed-unpaid bounty issue reports, we found 31 cases in which someone reminded the bounty hunter to claim the bounty, however, the reminder was ignored. By reassigning these unclaimed bounties to other issue reports, a larger value could be created for these “stale” bounties. For example, Bountysource can suggest and enable backers to assign their long-standing unclaimed bounties to another unaddressed issue report, which has many comments (i.e., people care about it), to encourage developers to address the issue report. Interestingly, we also found suggestions from developers who did not want to receive the bounty but suggested the bounty backers transferring the bounty to other issue reports or to the project as a kind of funding.

9 Threats to Validity

In this section, we discuss the threats to validity. Threats to **external validity** are related to the generalizability of our findings. We studied only bounty issue reports from GitHub and Bountysource. Further research is necessary to find whether our findings are generalizable to other types of issue reports (e.g., from commercial platforms) and other bounty platforms. Although our models have a high explanatory power, there might be additional factors that relate to the likelihood of an issue being addressed. Future studies should investigate more factors.

Threats to **internal validity** relate to the experimenter bias and errors. One threat relates to the project categorization in Section 5, in which we used 50 bounty issue reports as a threshold to distinguish whether a project uses bounties moderately or frequently. To alleviate this threat, we redid the analysis of Section 5 with other bounty-usage frequency thresholds (i.e., 40 and 60). The results show that our findings still hold (see our online appendix [26] for more details).

Another threat is that we rely on manual analysis to identify the addressed-unpaid issues and to identify why developers did not claim a bounty in Section 8.1, which may introduce bias due to human factors. To mitigate the threat of bias during the manual analysis, two of the authors conducted the manual analysis and discussed conflicts until a consensus was reached. We used Cohen’s kappa [5] to measure the inter-rater agreement and the value is 0.86, which indicates a high level of agreement.

10 Related work

In this section, we discuss related work along two dimensions: the bounty in software engineering and the improvement of the issue-addressing process.

Bounties in software development: Bounties are used to attract developers and motivate them to complete tasks. Prior work has studied the impact of bounties on software development. [11] gave an overview of bounties in Free/Libre/Open Source Software (FLOSS). They observed that bounty hunters’ responses are related to the workload, the probability of winning the bounty, the value of the bounty and the recognition that they might receive by winning the bounty. Different from their study, we focused on using bounties to improve the issue addressing process.

Several studies focused on the usage of bounties to motivate developers to detect software security vulnerabilities. Finifter et al. [4] analyzed vulnerability rewards programs for Chrome and Firefox. They found that the rewards programs for both projects are economically effective, compared to the cost of hiring full-time security researchers. Zhao et al. [23] investigated the characteristics of hunters in bug-bounty programs and found that the diversity of hunters improved the productivity of the vulnerability discovery process. Hata et al. [7] found that most hunters are not very active (i.e., they have only a few contributions). These findings are similar to our finding that most hunters only addressed one bounty issue. Zhao et al. and Maillart et al. [13], [24] analyzed the effect of different policies of bug-bounty programs. By studying bug-bounties from several perspectives, they provided insights on how to improve the bug-bounty programs. For example, Maillart et al. [13] suggested project managers to dynamically adjust the value of rewards according to the market situation (e.g., increase rewards when releasing a new version).

However, there is not much research to study the effectiveness of bounties in the issue-addressing process. The work of Kanda et al. [10] is closest to ours. They studied GitHub and Bountysource data but studied only 31 projects (compared to 1,203 in our study). They compared the closed-rate and closing-time between bounty issue and non-bounty issue reports. Their results showed that the closing-rate of bounty issue reports is lower than that of non-bounty issue reports, and it takes longer for the bounty issue reports to get closed than non-bounty issue reports. Our study performs a deeper analysis of bounties at the project and the time level. Besides, we further study the relationship between the issue-addressing likelihood and the bounty-related factors (e.g., the total bounty value of a bounty issue report) while controlling for the factors that are related to the issue report and project (e.g., the number of comments before the first bounty is proposed). We found that the timing and the bounty-usage frequency are the most important factors in increasing the issue-addressing likelihood.

Improving the issue-addressing process: Issue addressing is an essential activity in the life cycle of software development and maintenance. Therefore, a large amount of research was done to improve the issue-addressing process. One group of studies focused on providing insights into improving the issue-addressing process in aspects of the quality of issue reports, the effectiveness of developers and automated bug localization and fixing. For example, Bettenburg et al. [2], [8] analyzed the quality of bug reports (i.e., a type of issue report) and provided some guidelines for users to generate high-quality reports so that developers can address issues more efficiently. Ortu et al. [15] analyzed the relation between sentiment, emotions, and politeness of developers in comments with needed time to address an issue. They found that the happier developers are, the shorter the issue-addressing time is likely to be. Zhong et al. [25] performed an empirical study on real-world bug fixes to provide insights and guideline for improving the state-of-the-art of automated program repair. Soto et al. [19] performed a large-scale study of bug-fixing commits in Java projects and provided insights for high-quality automated software repair to target Java code. A number of studies helped developers locate the buggy code in projects using information retrieval techniques [18], [21], [22], [27]. Different from prior studies, we perform an empirical study to understand the relationship between bounties and the issue-addressing process. We provide insights into how to better use the bounty to improve the issue-addressing process.

11 Conclusion

In this paper, we studied 5,445 bounties with a total value of $406,425 from Bountysource along with their associated 3,509 issue reports from GitHub to study the relationship between the bounty (e.g., timing of proposing a bounty, bounty value, and bounty-usage frequency) and the issue-addressing likelihood. We found that:

1) The timing of proposing bounties and the bounty-usage frequency are the most important factors that impact the issue-addressing likelihood. Issue reports for which bounties were proposed earlier are more likely to have a higher issue-addressing likelihood and a faster addressing-speed.
2) In first-timer bounty-projects, the issue-addressing likelihood is higher for higher bounty values and in these projects, backers should consider proposing a relatively bigger bounty.
3) Backers should be cautious when proposing small bounties on long-standing issue reports as they risk losing money without getting their issue addressed.

Our findings suggest that backers should consider proposing a bounty early and be cautious when proposing small bounties on long-standing issue reports. Bounty platforms should allow dividing bounties between hunters, and transferring bounties to other issue reports.

References

[1] D. F. Bauer. Constructing confidence sets using rank statistics. *Journal of the American Statistical Association*, 67(339):687–690, 1972.

[2] N. Bettenburg, S. Just, A. Schröter, C. Weiss, R. Premraj, and T. Zimmermann. What makes a good bug report? In *Proc. of the Int'l Symp. on Foundations of Software Engineering*, pages 308–318. ACM, 2008.

[3] B. Efron. How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, 81(394):461–470, 1986.

[4] M. Finifter, D. Akhawe, and D. Wagner. An empirical study of vulnerability rewards programs. In *USENIX Security Symp.*, pages 273–288, 2013.

[5] K. Gwet et al. Inter-rater reliability: dependency on trait prevalence and marginal homogeneity. *Statistical Methods for Inter-Rater Reliability Assessment Series*, 2:1–9, 2002.

[6] F. E. Harrell, Jr. *Regression Modeling Strategies*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[7] H. Hata, M. Guo, and M. A. Babar. Understanding the heterogeneity of contributors in bug bounty programs. In *Proc. of the ACM/IEEE Int'l Symp. on Empirical Software Engineering and Measurement*, pages 223–228, 2017.

[8] P. Hooimeijer and W. Weimer. Modeling bug report quality. In *Proc. of the Int'l Conf. on Automated Software Engineering*, pages 34–43. ACM, 2007.

[9] S. Kabinna, C.-P. Bezemer, W. Shang, M. D. Syer, and A. E. Hassan. Examining the stability of logging statements. *Empirical Software Engineering*, 23(1):290–333, 2018.

[10] T. Kanda, M. Guo, H. Hata, and K. Matsumoto. Towards understanding an open-source bounty: Analysis of Bountysource. In *Int'l Conf. on Software Analysis, Evolution and Reengineering*, pages 577–578. IEEE, 2017.

[11] S. Krishnamurthy and A. K. Tripathi. Bounty programs in free/libre/open source software. In *The Economics of Open Source Software Development*, pages 165–183. Elsevier, 2006.

[12] J. D. Long, D. Feng, and N. Cliff. Ordinal analysis of behavioral data. *Handbook of psychology*, 2003.

[13] T. Maillart, M. Zhao, J. Grossklags, and J. Chuang. Given enough eyeballs, all bugs are shallow? Revisiting Eric Raymond with bug bounty programs. *Journal of Cybersecurity*, 3(2):81–90, 2017.

[14] S. McIntosh, Y. Kamei, B. Adams, and A. E. Hassan. An empirical study of the impact of modern code review practices on software quality. *Empirical Software Engineering*, 21(5):2146–2189, 2016.

[15] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli. Are bullies more productive?: Empirical study of affectiveness vs. issue fixing time. In *Proc. of the Working Conf. on Mining Software Repositories*, pages 303–313, 2015.

[16] G. K. Rajbahadur, S. Wang, Y. Kamei, and A. E. Hassan. The impact of using regression models to build defect classifiers. In *Proc. of the Int'l Conf. on Mining Software Repositories*, pages 135–145, 2017.

[17] J. Romano, J. D. Kromrey, J. Coraggio, and J. Skowronek. Appropriate statistics for ordinal level data: Should we really be using t-test and Cohen's d for evaluating group differences on the NSSE and other surveys. In *Annual meeting of the Florida Association of Institutional Research*, pages 1–33, 2006.

[18] R. K. Saha, M. Lease, S. Khurshid, and D. E. Perry. Improving bug localization using structured information retrieval. In *Proc. of the IEEE/ACM Int'l Conf. on Automated Software Engineering*, pages 345–355, 2013.

[19] M. Soto, F. Thung, C.-P. Wong, C. Le Goues, and D. Lo. A deeper look into bug fixes: Patterns, replacements, deletions, and additions. In *Proc. of the Int'l Conf. on Mining Software Repositories*, pages 512–515, 2016.

[20] S. Wang, T.-H. Chen, and A. E. Hassan. Understanding the factors for fast answers in technical Q&A websites. *Empirical Software Engineering*, pages 1–42, 2017.

[21] S. Wang and D. Lo. Amalgam+: Composing rich information sources for accurate bug localization. *Journal of Software: Evolution and Process*, 28(10):921–942, 2016.

[22] S. Wang, D. Lo, and J. Lawall. Compositional vector space models for improved bug localization. In *IEEE Int'l Conf. on Software Maintenance and Evolution*, pages 171–180, 2014.

[23] M. Zhao, J. Grossklags, and K. Chen. An exploratory study of white hat behaviors in a web vulnerability disclosure program. In *Proc. of the Workshop on Security Information Workers*, pages 51–58. ACM, 2014.

[24] M. Zhao, A. Laszka, and J. Grossklags. Devising effective policies for bug-bounty platforms and security vulnerability discovery. *Journal of Information Policy*, 7:372–418, 2017.

[25] H. Zhong and Z. Su. An empirical study on real bug fixes. In *Proc. of the Int'l Conf. on Software Engineering*, pages 913–923, 2015.

[26] J. Zhou. Supplementary material for our paper. https://github.com/yiikou/Bountysource/appendix.pdf, 2018.

[27] J. Zhou, H. Zhang, and D. Lo. Where should the bugs be fixed? - more accurate information retrieval-based bug localization based on bug reports. In *Proceedings of the 34th International Conference on Software Engineering*, pages 14–24, 2012.