Empirical Software Engineering (2022) 27: 24
https://doi.org/10.1007/s10664-021-10060-y

Studying donations and their expenses in open source projects: a case study of GitHub projects collecting donations through open collectives

Jiayuan Zhou[1] · Shaowei Wang[2] · Yasutaka Kamei[3] · Ahmed E. Hassan[4] · Naoyasu Ubayashi[3]

Accepted: 28 September 2021 / Published online: 30 November 2021

Abstract

Operating an open source project requires not only intrinsic motivation (e.g., the joy of participation) but also extrinsic motivation (e.g., financial incentives). Almost 95% of open source projects are no longer maintained after a year. Nowadays, although donations start to play an important role in operating open source projects, there is little knowledge about the characteristics of donors and the usage of donations. A better understanding of the characteristics of donations, their donors, and the usage of donations in open source projects is needed to provide insights to the stakeholders of open source projects to help them operate their projects more sustainably. In this paper, we study the donations that are received through the Open Collective platform (i.e., an online crowdfunding platform) to support open source projects, to understand the characteristics of these donations, their donors, and the usage of these donations. To do so, we investigate 225 GitHub open source projects that received 54,889 donations with a total value of $2,537,281 through the Open Collective platform. We find that: 1) In general, corporate donors tend to donate more money than individual donors in a single donation. However, in a collective, the total donation amount from individual donors is more than corporate donors, suggesting the importance of individual donors. Moreover, individual donors are more likely to redonate to the same collective compared to corporate donors. 2) Non-engineering-related expenses take up to 54.0% of the total number of all expenses that are filtered against donation. For instance, “Web services”, “marketing”, and “travel” expenses are the three most frequent and costly non-engineering-related expense types. For engineering-related expenses, the most frequent expenses are related to development and maintenance. Interestingly, we also observe that 18% of the engineering expenses were spent to propose bounties for addressing issues with a median cost of $95 per proposed bounty. We further analyze the differences between individual-supported collectives (i.e., collectives where more than 80% of their donation amount is

Communicated by: Maurizio Morisio

This work is not related to Jiayuan Zhou’s role at Huawei.

✉ Shaowei Wang
shaowei.wang@umanitoba.ca

Extended author information available on the last page of the article.

from individual donors) and corporate-supported collectives (i.e., collectives where more than 80% of their donation amount is from corporate donors). We observe that corporate-supported collectives tend to receive a higher donation amount than individual-supported collectives and the monthly received donation amounts are positively associated with the levels of community and maintenance activities in corporate-supported collectives. They have no significant difference in terms of popularity (e.g., the number of pull requests) of their associated GitHub projects. Our findings suggest that the stakeholders of GitHub open source projects should try to attract more individual donors. Collectives should not expect to receive a large amount of funds overall from donations unless their projects are very popular or are mainly supported by corporations. Projects should budget for a reasonable amount (e.g., 13% of total funds) of non-engineering expenses (e.g., marketing and traveling).

Keywords Donations · Open source · GitHub

1 Introduction

Open source projects are widely used by many companies, government agencies, and individuals. A prior study (Androutsellis-Theotokis et al. 2011) shows that 65% out of 1,313 surveyed companies relied on open source projects to expedite application development. However, operating open source projects is a challenging task. Operating open source projects (e.g., fixing bugs and maintaining documentation) requires a significant amount of effort from developers. However, “Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his project and distribute for free?”, as Bill Gates once noted.[1] In other words, we cannot expect all developers to be willing to volunteer for maintenance tasks (Steinmacher et al. 2018). As a result, 64% of well-known and popular open source projects rely on one or two contributors to manage most of their tasks (Avelino et al. 2016) and almost 95% of open source projects are no longer maintained after a year (Rich Sands 2012).

Financial incentives are an important extrinsic motivator for developers to sustain open source projects (Atiq and Tripathi 2016). More and more individuals and corporations make donations to open source projects. For instance, the Linux Foundation plans to provide 10 million dollars to support open source projects on Community Bridge.[2] More than 6 million dollars donations have been made through the *Open Collective* platform,[3] an online platform that hosts the funding for more than 500 open source projects.

Nowadays, donations play an important role in the smooth operation of open source projects.[4] However, little is known about the characteristics of donors and their donations, as well as how the received donations are spent. With a better understanding of such questions, we can provide insights to the stakeholders of open source projects to help them operate their projects. For instance, the stakeholders of an open source project can obtain deeper insights of the donations that they are likely to receive and the potential donors.

[1] https://genius.com/Bill-gates-an-open-letter-to-hobbyists-annotated

[2] https://www.linuxfoundation.org/press-release/2019/03/the-linux-foundation-launches-new-communitybridge-platform-to-help-sustain-open-source-communities/

[3] https://opencollective.com/

[4] https://opensource.guide/getting-paid/

A collective is a group of people sharing the same mission to collect donations. In this paper, we study 225 GitHub open source projects that set up collectives on the Open Collective platform for collecting donations. These collectives received 54,889 donations from 7,071 individual and 877 corporation donors, with a total value of $2,537,281. A total of 84.6% (i.e., $2,192,439) of the received donations have been spent on various operational activities (e.g., development and maintenance). We examine how donors make donations and how the received donations are spent to sustain open source projects. We found that:

1. In general, corporate donors tend to donate more money (with a median value of $25) than individual donors (with a median value of $5) in one donation. However, in a collective, the total donation amount from individual donors ($833 in median) is more than corporate donors ($550 in median), suggesting the importance of individual donors. Moreover, individual donors are more likely to redonate to the same collective compared to corporate donors.
2. Non-engineering-related expenses take up to 54% of the total number of all expenses. "Web services", "marketing", and "travel" expenses are the three most frequent and costly non-engineering-related expense types. For engineering-related expenses, the most frequent expenses are related to development and maintenance. Interestingly, we also observe that 18% of the engineering expenses were spent to propose bounties for addressing issues with a median cost of $95 per proposed bounty.

We further analyze the differences between individual-supported collectives (i.e., collectives where more than 80% of their received donation amount is from individual donors) and corporate-supported collectives (i.e., collectives where more than 80% of their received donation amount is from corporate donors). We observe that:

1. Corporate-supported collectives tend to receive a higher monthly and total donation amount than individual-supported collectives.
2. The monthly received donation amounts are positively associated with the levels of community and maintenance activities in corporate-supported collectives.
3. There is no significant difference between corporate-supported and individual-supported collectives in terms of the popularity of their associated GitHub projects and both collectives are likely to spend donations on a small group of specific types of expenses (e.g., engineering and web services).

Our findings suggest that the stakeholders of GitHub open source projects should try to attract more individual donors since they will donate more money overall and more steadily than corporate donors. Collectives should not expect to receive a large amount of funds overall from donations (e.g., over $10,000) unless their projects are very popular (e.g., more than 9,000 issue reports) and are mainly supported by corporations. Projects should budget for a reasonable amount (e.g., 13% of total funds) of non-engineering expenses (e.g., marketing and traveling).

Paper Organization Section 2 presents background information about the Open Collective platform. Section 3 describes our data collection process. Sections 4 and 5 study donations in terms of donors, donations and expenses. Section 6 studies the differences between individual-supported and corporate-supported collectives. Section 7 studies interesting examples of donations and discusses the implications of our study. Section 8 discusses threats to the validity of the observations of our study. Section 9 introduces related work. Finally, Section 10 concludes our study.

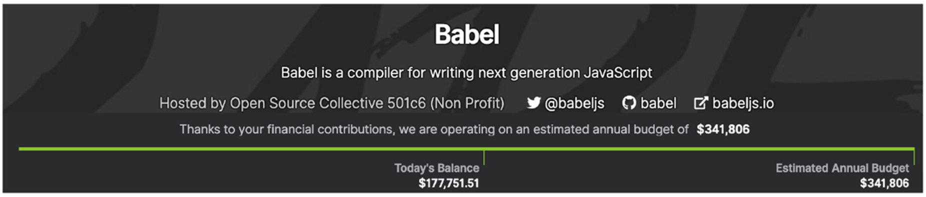


Fig. 1 An example of the collective of an open source project

2 Background

Open Collective,[5] *Patron*,[6] and *Salt*[7] are examples of online platforms for crowdfunding to support the operation of open source projects. The Open Collective platform is one of the most popular platforms (Overney et al. 2020). The platform facilitates a transparent mechanism for managing donations tracking their usage (i.e., expenses), which enables each donation and expense transaction transparent. In other words, it enables us to collect various information about donations, e.g., the corresponding donors and the usage of these donations. Hence, we study the donations that are collected on the Open Collective platform.

We provide below a brief background of the Open Collective platform along the following three dimensions: collective, donor, and how the platform works.

Collective A collective represents a group of people sharing the same mission or purpose and they raise and spend funds transparently to achieve. People can set up their collectives on the Open Collective platform for collecting donations without charge. There are many types of collectives according to their missions, such as ones for supporting open source projects, meetups, and non-profits organizations. For example, the `WWCode Toronto` collective[8] is to support a non-profit organization aiming at inspiring women to succeed in technology careers, and the `Babel` collective[9] is created to support the `Babel` open source project. Collectives that are associated with open source projects can choose to add their associated GitHub repository link to the official homepage of their Open Collective (see Fig. 1). We focus on collectives that are associated with open source projects. We introduce how we identify such collectives in Section 3.

Members of a collective can submit their expenses to a collective for their contributions or for reimbursements. When an expense is submitted, the expense will be labeled with a specific expense type to represent the main purpose of the expense. Unfortunately there is no uniform definition for expense types across collectives. Hence, we manually relabeled the expense types. We elaborate on our relabeling process in Section 5. After an expense is submitted, administrators of the collective receive a notification and they need to decide whether to approve or reject the expense.

[5] https://www.opencollective.com
[6] https://www.patreon.com
[7] https://salt.bountysource.com
[8] https://opencollective.com/wwcodetoronto
[9] https://opencollective.com/babel

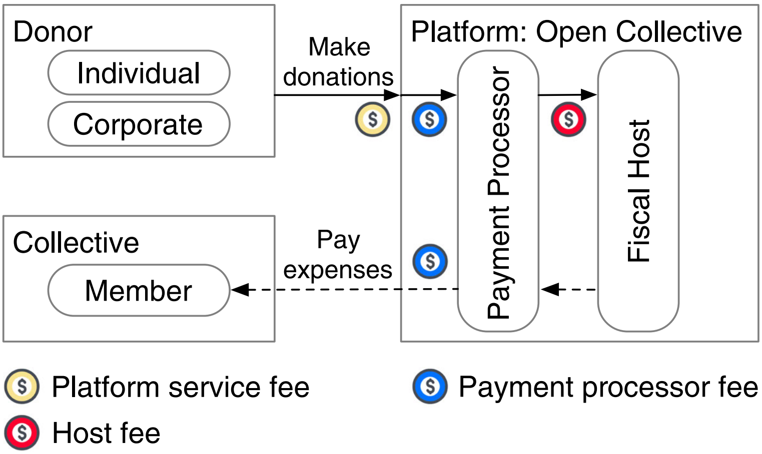


Fig. 2 The transaction flow for donating or paying an expense on the Open Collective platform

Donor There are two types of donors: *individual* and *corporate* donors.[10] Donors can choose to make a one-time donation or recurring donations. Open Collective platform provides two automatically recurring donation frequency options (i.e., month-based and year-based) to donors. We refer to these three different donation styles as the one-off, monthly, and yearly donation styles, respectively.

Donations can be made through a credit card, a gift card, or the balance of their collectives or organizations. Since October 06, 2017, donors can attach a brief message to their donations to explain the rationale for their donations.

The Workflow of the Open Collective Platform The Open Collective platform plays the role of an agency between collectives and donors. The platform provides a payment processing service through several payment processors, such as *Stripe*[11] and *Paypal*,[12] so that donors can make donations conveniently. The platform also provides a fiscal sponsorship service by connecting several fiscal hosts, which help collectives to store their funds, generate invoices, and pay expenses, so that collectives do not need to create their own legal entity and bank account.

Figure 2 shows the transaction flows of donations and expenses. The colorful coins refer to the different types of fees that are charged for different transactions. When a donor makes one donation, the platform will charge a 5% service fee (i.e., the yellow coin) then the funds will go through a payment processor to a fiscal host. A payment processor will charge a payment processing fee (i.e., the blue coin), which is around 3%. After that, the fiscal host will host the funds and charge a hosting fee (i.e., the red coin). The hosting fee varies across fiscal hosts. For example, the `Open Collective Foundation` fiscal can host US-based charity projects and the fee is 5% for each donation.[13] When a submitted expense is approved, the expense will be paid from the platform to the expense submitter. During this transaction, only the payment processing fee will be charged. In general, the total service fee

[10]https://docs.opencollective.com/help/about/terminology
[11]https://www.stripe.com
[12]https://www.paypal.com
[13]https://docs.opencollective.com/help/hosts

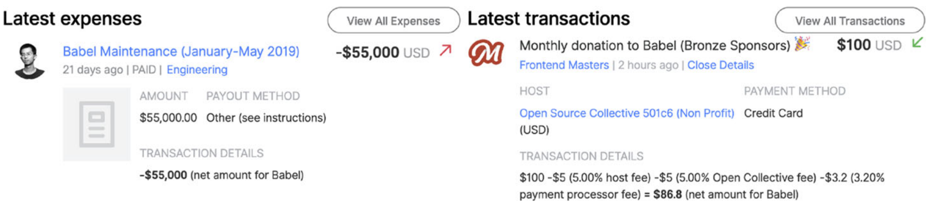


Fig. 3 An example of transaction record from the *Babel* collective

for a donation or an expense ranges from 8% to 13% of the total amount of transaction.[14] For example, Fig. 3 shows an expense transaction and a donation transaction along with their corresponding fees in detail.

The platform supports eight currencies (i.e., USD, EUR, CAD, MXN, GBP, AUD, NZD, and JPY)[15] and all transactions (i.e., donations and expenses) within the platform are visible to the public.

3 Data Collection and Experimental Setting

The Open Collective platform publishes its dataset[16] in the CSV format. The dataset consists of 804 collectives and their donation and expense records ranging from Jan. 23, 2015 to Jan. 31, 2019. The dataset also includes the donation messages from Oct. 06, 2017, to Apr. 12, 2019.

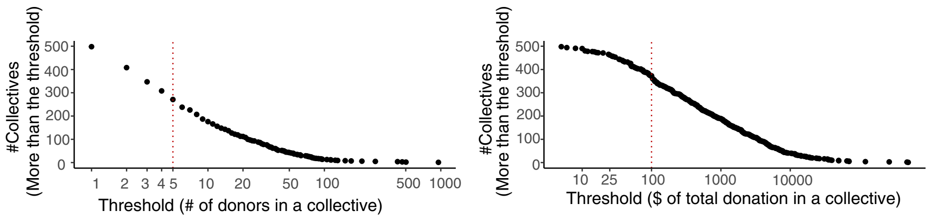
Because we wish to study donations and expenses that are for open source projects that are hosted on GitHub, we only focus on the collectives that are associated with open source projects. In other words, we select our studied collectives based on the following two criteria: 1) the collectives are for open source projects; 2) the collectives are hosted on GitHub. To do so, we automatically extracted 418 (out of 818) collectives that have GitHub repository addresses in their descriptions by searching for the keyword “github.com/”. To ensure that all the studied repositories are software development repositories, the first two authors further examined the description of these 418 GitHub repositories manually. We observe that four out of these 418 repositories are not for software development. We did not consider these four repositories in the rest of our analysis. For the rest 386 (out of 818) collectives that do not provide GitHub repository addresses in their descriptions, we checked for their GitHub repository by manually searching the names of these collectives through Google and GitHub’s search engine. If a GitHub repository exists for a collective, we collected the address for that collective. After manually verifying, we found another 102 collectives that have GitHub repositories. In total, we collected 516 collectives that are associated with open source projects.

We observe that some collectives receive a large total donation amount of donations from a small number of donors. We also observe that some collectives only received very few donations in total. For example, the `docker.io` collective only received one donation with an amount of $6. To reduce the bias that is introduced by collectives with a small number of donors or a low total donation amount, we use a threshold-based approach to further

[14] https://docs.opencollective.com/help/collectives#how-much-does-it-cost

[15] https://docs.opencollective.com/help/product/currencies

[16] http://drive.opencollective.com



(a) (b)

Fig. 4 The number of collectives under different thresholds of (a) donor number and (b) total donation amount in a collective

select proper collectives. We chose the number of donors and the total donation amount in a collective as thresholds. Figure 4a and b present the number of collectives under different thresholds. We empirically selected collectives with more than 5 donors and more than $100 total donation amount, to study collectives that leverage the donation system.

Since donors can make donations in six different currencies, we convert all currencies into United States Dollar (USD) using the daily exchange rate provided by OANDA to better conduct our study in terms of donation and expense amounts.[17]

Our experiments are conducted on Ubuntu 16.04 64 bits, with an E7- 4870 @ 2.40GHz CPU. All experiments are conducted using R studio data analysis tool (version: 1.2.1335).[18] We used the *cor*, *wilcox.test* and *cliff.delta* function in the stats `R` package (version 3.6.2) to compute Spearman's rank correlation coefficient, Wilcoxon rank-sum test, and Cliff's delta d effect size, respectively.

Results Finally, our studied dataset contains 225 collectives, 7,446 donors, 54,889 donations with $2,537,281 donation amount, and 1,626 expenses with a total value of $1,592,301. The median donation amount is $5 and the median expense amount is $135. The median number of contributors of collectives is 273 and the median size of collectives is 99.52 MB. Tables 1 and 2 give an overview of our dataset. Figure 5 shows the frequency of the major programming language of collectives. We observe that JavaScript is the most popular programming language among collectives.

4 RQ1: What are the Characteristics of Donors and Their Donations?

Motivation As we introduced in Section 2, there are two types of donors—individual and corporate donors. A corporate donor represents a legal entity instead of an individual. Due to their different nature, these two types of donors may exhibit different donation characteristics. For example, a corporate donor may make donations more frequently with higher amounts than an individual donor. Additionally, it is interesting to know the characteristics of donations within a collective. For instance, what is the proportion of the donations that are contributed by these two types of donors, and whether donors across these two types

[17] https://docs.opencollective.com/help/product/currencies,
https://www.oanda.com/fx-for-business/exchange-rates-api/daily-average-exchange-rates
[18] https://www.rstudio.com/

Table 1 Dataset description

Period	Nov. 23, 2015 to Jan. 31, 2019
Number of collectives	225
Number of expenses	1,626
Total amount of expenses	$1,592,301
Number of donations	54,324
Total amount of donations	$2,013,010
Number of donors	7,396
Number of donation messages	589

of donors tend to redonate to the same collective. With a better understanding of the characteristics of donors and their donations, the stakeholders (e.g., operators) of a collective can have a better understanding of the donations that they would typically receive and the potential types of donors that they might be able or wish to attract.

Approach First, we compare the donation amount in terms of different donation styles (i.e., one-off, monthly, or yearly donations), then we compare the amounts and frequencies of donations that are made by individual and corporate donors among all collectives. We employ the Wilcoxon rank-sum test (Bauer 1972) to measure whether or not the differences between individual and corporate donors are statistically significant. We calculate Cliff's delta d effect size (Long et al. 2003) to quantify the magnitude of the differences of the amount and frequency of donations between the two types of donors.

To further evaluate the likelihood of a donor redonating to a collective, we employ the *sticky* metric from a prior study (Yamashita et al. 2016). The value of the *sticky* metric reflects the proportion of donors that donated for a collective in the prior period (e.g., 6 months) and re-donate in the current period (e.g., recent 6 months) for the same collective. We refer such donors as to sticky donors of the collective.

Then we study donors and their donations at the collective level. To do so, we first calculate the proportion of individual and corporate donors within each collective, and compare the proportion of individual and corporate donors across collectives. We also calculate the proportion of the total amount of the donations that were contributed by these two types of donors in each collective, and compare the distributions of donation amount proportion across collectives. We use the Wilcoxon rank-sum test and Cliff's delta d effect size to measure the significance and magnitude of distribution of these two types of donors.

We further conducted a stratified analysis for collectives by considering the activeness and community size of their associated projects. We chose the total number of commits in the projects that are associated with a collective, as a proxy of the activeness of a collective.

Table 2 The five-number summary of the donation amount, the expense amount, the number of collective contributors, and collective size

	Quantile				
	Min.	1st.	Median	3rd.	Max.
$Donation amount	1	2	5	10	25,000
$Expense amount	1	45	135	514	100,000
#Collective contributor	1	110	273	800	23,923
Collective size (MB)	0.01	35.92	99.52	345.94	9629.30

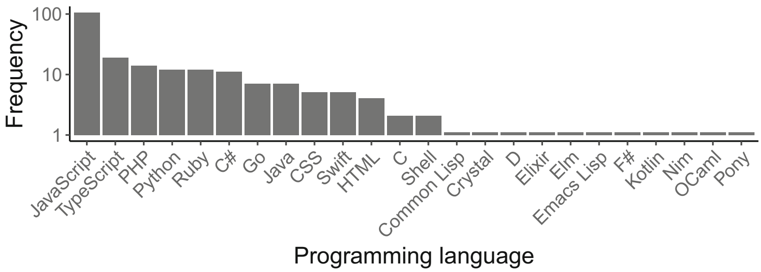


Fig. 5 The frequency of the major programming language of the studied collectives

We chose the total number of stars, watches, and forks of the projects that are associated with a collective as a proxy of the community size of the collective. Since the numbers of stars, forks, and watches are highly correlated with each other (i.e., Spearman's correlation values range from 0.73 to 0.89), we chose the number of stars as a proxy of the community size in our stratified analysis.

Results **Donors tend to donate more money in a single donation with the one-off style** Most donations are donated in either "one-off" or "monthly" style and only 0.1% (73) of donations are yearly donations. Figure 6 shows the boxplot of donation amount for different donation styles. The median donation amount for "yearly", "monthly", and "one-off" is \$50, \$5 and \$20, respectively. We perform the Wilcoxon rank-sum test and Cliff's delta test to measure the differences between distributions of donation amount for "one-off" and "monthly" donation styles. The result shows that two distributions are significantly different (p-value < 0.05) with a large effect size (Cliff's delta $= -0.21$), indicating that for a single donation, donors tend to donate more in the one-off style than in the monthly style. We also measure the difference between distributions of donation amount for "yearly" and the other two donation styles and we observe significant differences (p-value < 0.05) with a small (Cliff's delta of 0.32) and a large (Cliff's delta of 0.75) effect size, respectively.

Corporate donors tend to donate significantly more money (a median of \$25) in a single donation than individual donors (a median of \$5). Figure 7 shows distributions of donation amount for individual donors and corporate donors. Corporate donors made donations with a median amount of \$25, which is five times that of the individual donors' median amount. The Wilcoxon rank-sum test shows that there exists a significant difference

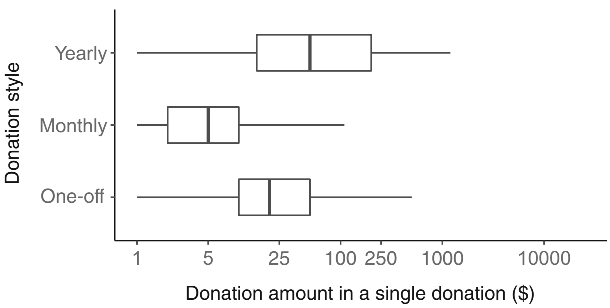


Fig. 6 The boxplot of donation amounts for different donation styles. Note that for the yearly and monthly donations, we consider the amount of every single donation rather than the sum of all donations

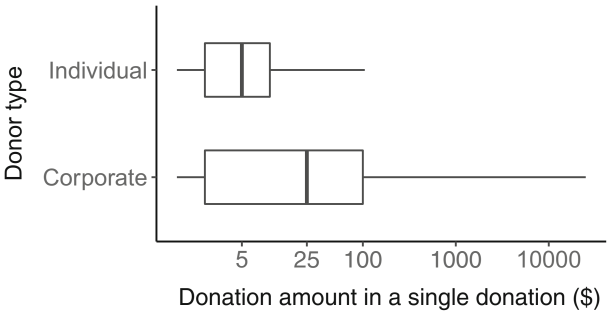


Fig. 7 The distribution of donation amount for different donor types

(p-value < 0.05) between these two distributions with a medium effect size (Cliff's delta $= -0.41$). The largest donation was made in Jan. 17, 2019, with an amount of $250,000 by the corporate donor *Modus Create*, which is a company aiming at digital transformation such as cloud migration. The median of the donation frequency for individual donors and corporate donors are both three. There is no significant difference between the donation frequency distributions for individual and corporate donors.

There are significantly more individual donors (a median number of 14) than corporate donors (a median number of 3) in a collective. In general, the total donation amount from individual donors are significantly higher than that from corporate donors in a collective. Figure 8 shows the distributions of the proportion for individual and corporate donors in a collective. It is obvious that the number of individual donors is more than corporate donors in a collective. The median proportion of individual donors across all the studied collectives is 85% and that of corporate donors is 15%. The results of the statistical test show that individual donors are significantly more than corporate donors with a large effect size in one collective.

Figure 9 shows the distributions of the proportion of total donation amounts from individual and corporate donors in a collective. The median proportion from individual donors is 63% and that of a corporate donors is 33%. Across collectives, the median donation amount from individual and corporate donors are $833 and $550, respectively. Our observation highlights the *importance of individual donors*. Although the donation amount of an individual donor is less than a corporate donor, the total contribution from individual donors is significantly more than corporate donors. For example, 94.8% (3,917 out of 4,132) total

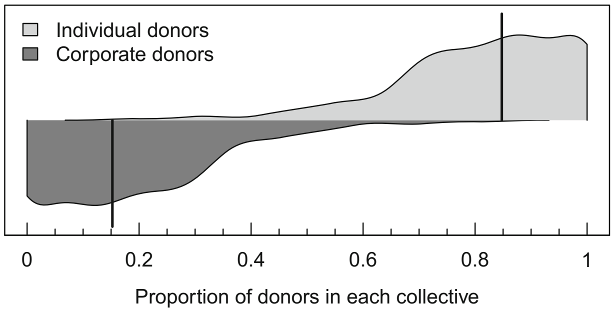


Fig. 8 The distribution of the proportion for the individual and corporate donors across collectives

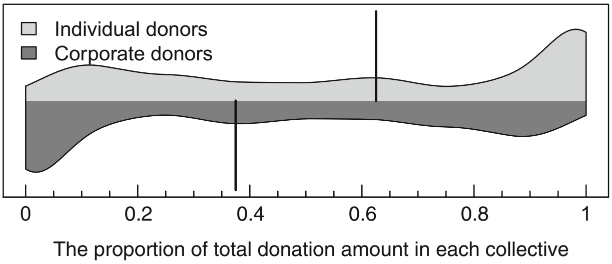


Fig. 9 The distribution of the proportion of the donation amount from individual donors and corporate donors across collectives

donation amount in the *ImageSharp*[19] collective is from individual donors and the median donation amount of the individual donors is \$5, which is smaller than the donation amount for corporate donors (\$20).

Figure 10a and b show the distributions of the proportion of the donation amounts between individual donors and corporate donors across collectives, under the two selected strata, #commit and #star, respectively. In Fig. 10a, we observe that for collectives that have no more than 15k commits, the proportion of donations from individual donors is more likely to be larger in the collectives that have more commits. However, for collectives that have more than 15k commits, the median proportion of total donation amounts from corporate donors is close to that of individual donors, probably indicating that corporate donors are more interested in supporting collectives that are highly active (e.g., more than 15,000 commits). Similarly, in Fig. 10b, we observe that in general, corporate donors tend to donate a larger amount of donations than individual donors in collectives that have more stars (e.g., more than 15,000 stars).

Individual donors are more likely to continue to redonate to the same collective to which they previously donated compared to corporate donors. Figure 11 shows the distributions of the sticky value of individual donors and corporate donors. The median sticky value for individual donors is 0.60, while that of corporate donors is 0.33. The Wilcoxon rank-sum test shows that there exists a significant difference (p-value < 0.05) between these two distributions with a small effect size (Cliff's delta $= 0.17$), indicating individual donors are more likely to redonate than corporate donors. In addition, to understand if sticky donors are usually active in making donations, we examine the frequency of donations for sticky donors. Figure 12 shows the donation frequency of sticky individual donors and sticky corporate donors. In median, the sticky individual donors donate 10.5 times and sticky corporate donors donate 9.5 times, which is much higher than the average level of donors, i.e., 4 and 5.2 for individual and corporate donors, respectively. Figure 13a and b show the distributions of sticky value for individual and corporate donors for each collective, under two selected metrics #commits and #stars, respectively. We observe that in general individual donors are more sticky to collectives than corporate donors, except for the collectives with more than 15,000 stars.

[19] https://opencollective.com/imagesharp

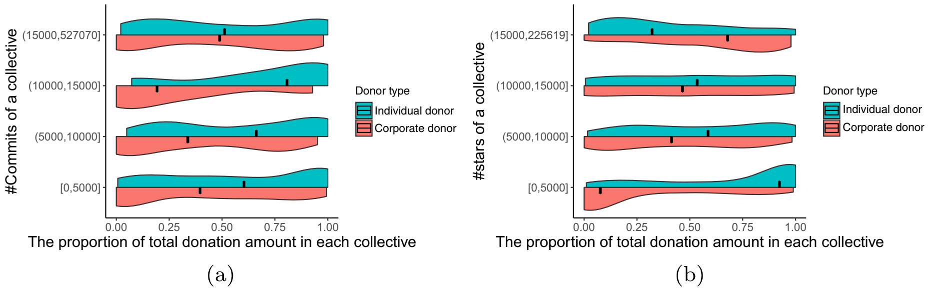


Fig. 10 The the proportion of the donation amount between individual donors and corporate donors across collectives, under two selected metrics, (a) #commits and (b) #stars, respectively

In general, corporate donors tend to donate larger amounts (with a median value of $25) than individual donors (with a median value of $5). However, in a collective, the total donation amount from individual donors ($833 in median) is more than corporate donors ($550 in median), which highlights the importance of individual donors (i.e., the value of working to attract more individual donors to one collective) Moreover, individual donors are more likely to redonate to a collective than corporate donors.

5 RQ2: What are the Received Donations Spent on?

Motivation Operating (i.e., developing and maintaining) open source projects encounters various types of expenses (e.g., development cost and website hosting cost). It is challenging to estimate the various expenses for operating an open source project.[20] Therefore, a study of the types of expenses for operating open source projects would be of great value to the leader and stakeholders of open source projects. In this RQ, we first provide an overview of the types of expenses across collectives. Then we further analyze the non-engineering-related and engineering-related expenses, respectively. A better understanding of the cost of operating such projects can help open source project stakeholders understand their budgets more sensibly.

5.1 Overview of Engineering-Related and Non-engineering-related Expenses

5.1.1 Approach

To provide an overview of the types of expenses across collectives, we first calculate the number of collectives that have expenses and no expense, respectively. Then we further compare the monthly expense amount across collectives between two families of expenses—non-engineering-related expenses and engineering-related expenses. We use the median value of the total expense amount of each month to represent the monthly expense amount of a collective.

[20] https://thenewstack.io/survey-open-source-programs-are-a-best-practice-among-large-companies/

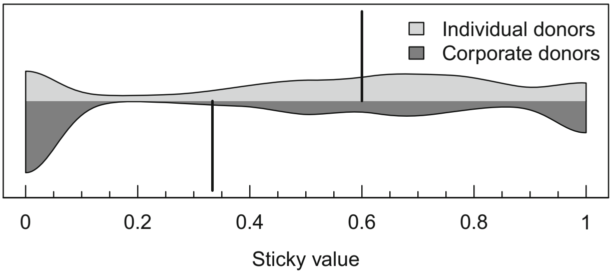


Fig. 11 The distribution of sticky value for individual and corporate donors for each collective

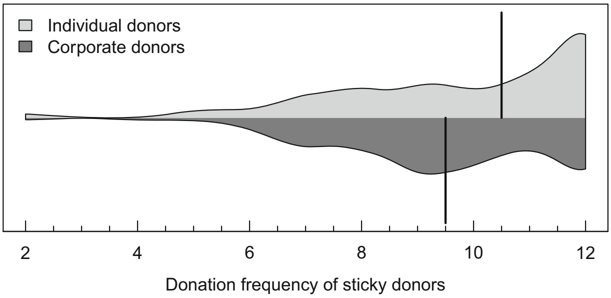


Fig. 12 The distribution of donation frequency for individual and corporate donors who are sticky to a collective

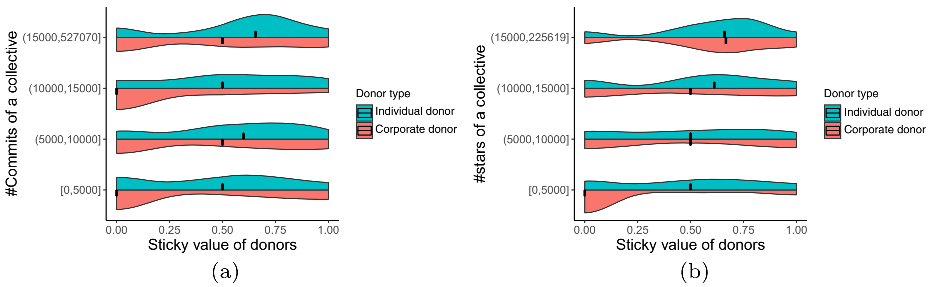


Fig. 13 The distributions of the proportion of the donation amount from individual donors and corporate donors across collectives, under two selected metrics, (a) #commits and (b) #stars, respectively

To identify the engineering-related and non-engineering-related expenses, we manually examine the expenses and the labels of each expense. These labels are provided by users when submitting expenses. In total, there are 139 (out of 225) collections that have expenses. We observe 15 expense labels (name, the food, coverage, supplies and material, office, team, design, web services, engineering, marketing, communications, travel, donation, legal, fund, and other) that are tagged by users. After examining the expenses and their labels, we observe some original expenses are labeled inaccurately due to any of the following three reasons:

1. **Expenses have different labels, while their purposes are similar.** For example, the description of an "office" type expense is "mac usb hub", and the description of a "supplies and materials" type expenses is "hardware renewal". However, the purposes of both expenses are the same, buying supplies and materials. Therefore, for consistency, these two expenses should be tagged with the same label.
2. **Expenses have the same label, while purposes are different.** For example, the descriptions of three "communication" type expenses are "travel to Poland sprint" (i.e., onsite meeting), "MailChimp email service" (i.e., the web communication service), and "community maintenance" (e.g., triage issues). However, "travel to Poland sprint" is for traveling, "MailChimp email service" is for SaaS, a type of web service, and "community maintenance" relates to maintenance. For consistency, these three types of expense labels should be labeled with three labels.
3. **Expenses have the wrong labels.** We also observe that some expenses were labeled with wrong labels. For example, the expense with a description "project maintenance and enhancement" should be labeled with "engineering" rather than "design".

In order to reduce the bias from these inaccurate labels when analyzing different types of expenses, the first and the third authors manually relabeled all expenses using the existing original expense types provided by users. Note that we merged type "office" type into type "supplies and materials" since they share a similar purpose. For the "communication" type expenses, we split and merged them into eight other expense types, respectively. Table 3 shows the merged result of 124 communication type expenses. For example, the expenses that were for the onsite meeting (e.g., "Berlin Meetup Organizer Costs") were merged into "travel" type expenses, the expenses that related to web communication services were merged into "Web services" type expenses, and the expenses that were for maintenance were

Table 3 The results of re-labeled 124 communication expenses and their corresponding examples of expense descriptions

Re-labeled type	Relabeled percentage	Total amount($)	Examples of expense descriptions
Engineering	45%	73,975	"Reviewing issues and pull requests."
Web services	27%	2,461	"Mail chimp email list."
Marketing	12%	4,309	"Ordering goodies (stickers, mugs ...) for contributors and upcoming events."
Travel	6%	3,229	"Travel to poland sprint."
Other	4%	1,612	"July invoice."
Team	1%	53	"Hoodie."
Food & Beverage	1%	70	"Breakfast and snacks for national day of civic hacking."
Supplies & materials	1%	92	"Power strips and extension cord."

merged into "engineering" type expenses. Then we removed the "communication" expense type. We consider all expenses of type "Engineering" as engineering-related expenses and the rest of types of expenses as the non-engineering-related expenses.

Table 4 The different types of expenses along with corresponding examples

Expense type	Explanation of expenses	Examples from actual expense descriptions
Engineering	Implementing new features, addressing reported bugs, and maintenance related costs.	"App development in October", "Community maintenance", and "April 2018: Documentation updates".
Web services	Web hosting and SaaS (i.e., Software as a service) subscription costs.	"GoDaddy domain name cost", "Heroku Hosting costs for July 2015", and "Canny.io annual subscription".
Website appearance design	Expenses for website appearance design (e.g., icons or theme) related costs.	"Roots Website Redesign (Concept, Colors, First Designs)", "Open Source Design Rollup-Banner", and "Theme/logo design for forum".
Donation	Donating to other collectives.	"Donation to the Python Software Foundation", and "Donation expanse for obfuscator.io domain".
Food & Beverage	Food and drink expenses for meetings or events.	"Food for the team meeting in Amsterdam", "Pizza for PDXNode Hack Night".
Legal	Bookkeeping, accounting, and brand registration costs.	"Brand registration", "Watson & Associates - Bookkeeping - March 2018", "Watson & Associates - Quarterly Accounting - January 2018".
Marketing	Advertisement and related costs (e.g., stickers, business cards) for attracting more users.	"New Logo Design", "Stickers for the conference", and "Printing signage and business cards".
Travel	Meeting and attending events (e.g., conferences) related costs (e.g., transportation, accommodation).	"Train for Vue.js conference", "Conference travel reimbursements for Q4 2018", and "Airbnb for Vue Sprint in Poland".
Team	Expenses for team activities (e.g., team t-shirts and video games).	"Core team T-shirts", "Mailing custom t-shirts to contributors", and "destiny 2 digital deluxe edition".
Supplies & Materials & Office	Office supplies and equipment costs.	"Mac USB hub", "Postage and Envelopes", and "Raspberry Pi Zero W + Case".
Other	Others expenses.	"Emergency expenses", "Portuguese translations from Urb-i", and "Transfering collective to EU host".

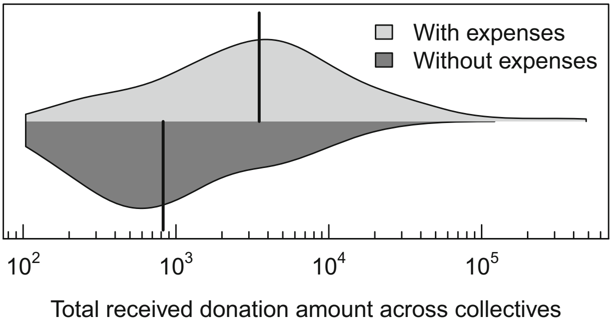


Fig. 14 The distribution of total received expense amount for the collectives with expenses and the ones without expenses

5.1.2 Results

After manually relabeling all expenses, we ended up with 11 expense types. Table 4 shows the explanations and examples for these expense types. Cohen's Kappa is 0.91, which indicates a high level of inter-rater agreement.

Overall, 38.2% (86 out of 225) collectives have no expenses. The possible explanation is that such collectives do not receive enough donations to pay the expenses. Figure 14 shows the distributions of the total received donation amount for the collectives that have expenses and ones without expenses. The median amount of the received donations for the collectives without expenses and collectives with expenses are \$824.5 and \$3,504, respectively. The Wilcoxon rank-sum test shows that the difference between these two distributions is significantly different (p-value < 0.05) with a medium effect size (Cliff's delta $= 0.47$).

Non-engineering-related expenses occur more frequently than engineering-related expenses. However, the amount of engineering-related expenses are higher than non-engineering-related expenses. 75.0% (104 out of 139) of the collectives with expenses have non-engineering-related expenses, which is higher than the proportion of collectives that have engineering-related expenses (55.4%). Such non-engineering-related expenses take up 45.6% (i.e., 665 out of 1,459) of all expenses. In terms of the expense amount across

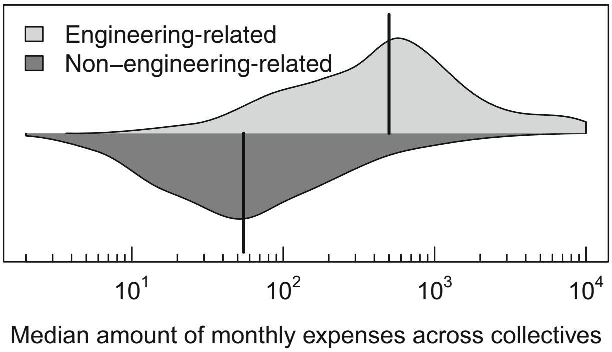


Fig. 15 The distribution of collectives' median monthly expense amount that were used for engineering-related versus non-engineering-related expenses

all collectives, 87.0% (median) of the total expense amount is spent on engineering-related expenses and 13.0% (median) of the total expense amount is spent on non-engineering-related expenses. Figure 15 shows the distribution of median monthly expense amount of a collective for engineering-related and non-engineering-related expenses across collectives. The median amount of engineering-related expenses is \$500 and that of non-engineering-related expenses is \$54.75. The Wilcoxon rank-sum test shows that the difference between these two distributions is significantly different (p-value < 0.05) with a large effect size (Cliff's delta $= 0.71$), indicating that collectives spent significantly more funds on engineering-related expenses than non-engineering-related expenses.

We perform further analysis on the engineering-related and non-engineering-related expenses and elaborate on the results in the following sections.

5.2 Non-engineering-related Expenses

5.2.1 Approach

To study how collectives spent money on non-engineering-related expenses, we first analyze how frequently a non-engineering-related expense type is spent across collectives. For each non-engineering-related expense type, we calculate the number of collectives that have ever spent money on this type. Then for each collective, we calculate the proportion of the amount of each type of non-engineering-related expenses.

To further understand the purpose of non-engineering-related expenses and how widely such purpose is applied across collectives, we calculate the frequency of words appearing in expense descriptions. More specifically, we count a word only once even if it appears more than one time in the description of a collective.

We perform preprocessing on the raw description of expenses before analyzing them. We perform tokenization, stemming, and stop word removal on the raw description of each expense. We use the **tokenizers**[21] `R` package for tokenization and stemming. To remove stop words, we not only remove the stop words listed in the **stopwords**[22] `R` package, but also consider the collective names as stop words and remove them. To reduce any potential bias due to the synonym words, we also replace synonyms or short forms manually. For instance, we replace "development engineering" with "development" and "bounty program" with "bounty".

5.2.2 Results

"Web services", "marketing", and "travel" are the three most frequent and costly expense types among the nine non-engineering-related expense types. Different collectives use their received donations for different purposes. For example, the *Storybook* collective used all received donations on marketing. Figure 16a shows that "web services", "marketing", "travel", and "supplies & materials" are the four most widely used expense types. Especially, 29% (65) collectives have "web services" expenses and 19% (43) collectives have "marketing" expenses. Figure 16b shows the distribution of cost proportions of non-engineering-related types of expenses across collectives. We observe that "web services", "marketing", and "travel" still are the top three non-engineering-related expense

[21] https://cran.r-project.org/web/packages/tokenizers/index.html

[22] https://cran.r-project.org/web/packages/stopwords/index.html

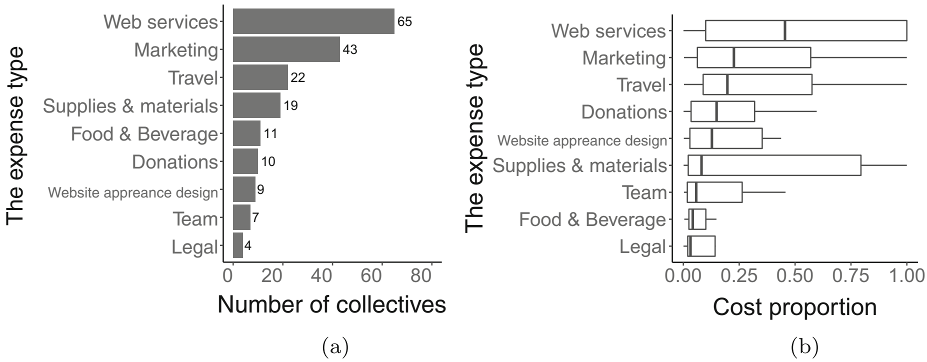


Fig. 16 (a) The number of collectives that have each non-engineering-related expense types. (b) The distribution of cost proportion of each non-engineering-related expense types across collectives

types with the highest median cost proportion. However, the median cost proportion of "supplies & materials" is low (i.e., 5%). We observe that among 19 collectives, more than half of them only have 1 expense for supplies and materials and the median expense amount is \$75.5, which normally takes up a small cost proportion in these collectives. By manually checking the corresponding expense descriptions, we observe that 41 of them are related to hardware (e.g., a laptop for development) and accessories (e.g., SD cards for testing server hardware) for test or development purposes. Overall, "web services", "marketing", and "travel" are the three most common and costly non-engineering-related expense types.

Figure 17 shows the collective-level frequency of the top five frequent words for "web services", "marketing", and "travel" expenses, respectively. From the figure, we observe some possible purposes for "web service" expenses such as the registration or renewal of domain name, service, server hosting, or digital license renewal. Possible purposes for "marketing" expenses were the making stickers (from stickermule.com), designing logos, or hosting onsite events (e.g., meetups). Interestingly, 47% of marketing expenses are for

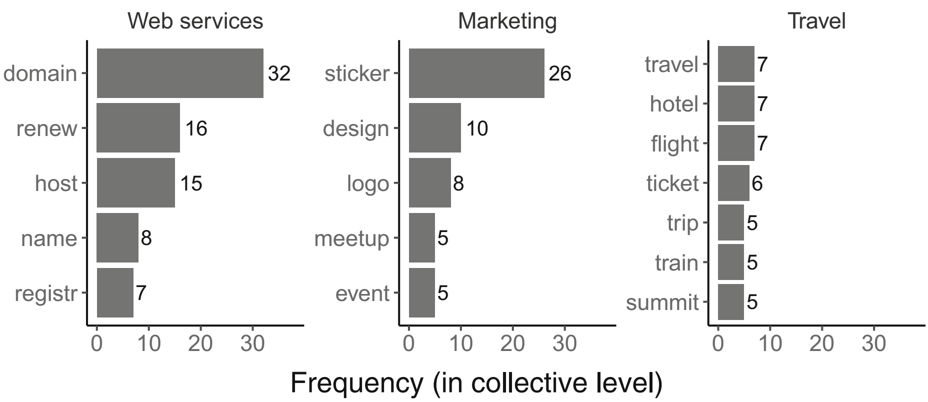


Fig. 17 The frequency of the top five frequent keywords for "web services", "marketing", and "travel" expenses, respectively, at the collective level (i.e., the frequency of a keyword is counted once for each collective)

making stickers. Possible purposes for "travel" expenses were for developer on offsite activities (e.g., conferences and summits), transportation fees (e.g., flight, train, and lyft), and accommodation fees (e.g., hotel).

5.3 Engineering-Related Expenses

5.3.1 Approach

To study the purpose of engineering-related expenses, we applied the same text preprocessing process on the expense descriptions. We calculate the word frequency for the words of expense descriptions. We assume that such frequent words reflect the purposes of engineering expenses. Table 5 presents the top 10 most frequent words (in stemmed form) appearing in the engineering-related expenses. We notice that except for the stemmed words "support" and "contribut" (marked with *) that are vague, the other eight stemmed words represent a software engineering task. Note that we consider an expense to involve a software engineering task if the expense description contains task-related words. For example, if an expense description contains the frequent words "development" and "documentation", we consider this expense to involve two tasks that are related to development and documentation.

In some cases, one expense could be associated with several purposes (i.e., one expense description may contain more than one of the frequent words in Table 5) and we do not know the cost portion for each purpose. For example, we cannot estimate the cost of maintenance and development cost for a $3,500 expense, with the description that says "Maintenance & Development 10/2017". Hence, we focus on expenses that have only one single purpose (i.e., only contain one of the frequent words that are listed in Table 5) when analyzing the cost of a specific purpose of an expense. For example, for a $100 expense, with the description that says "Webpack development", we consider the development cost to be $100. Table 6 presents the frequency and five-number summary of the expense amount for the most frequent eight software engineering tasks. Note that we cannot estimate the expense amount for the "communication" task since it is always mentioned with other tasks in the same expense description.

Table 5 The top 10 most frequent stemmed words and a corresponding example of expense for each of these words

Freq. (%)	Stemmed word	Examples of expense descriptions that contain the stemmed word
268 (40.3%)	develop	"App development in October".
97 (14.6%)	mainten	"Community Maintenance".
72 (10.8%)	contribut*	"Contribution to webpack".
68 (10.2%)	bounti	"Bug Bounty claim $100".
55 (8.1%)	issu	"Work on PR #805 (issue #787)".
51 (7.7%)	document	"Documentation on webpack".
40 (5.8%)	support*	"General Support".
33 (5.0%)	communic	"Development and Communication"
24 (3.6%)	releas	"v0.19.0 Release"
16 (2.3%)	test	"JHipster VueJS - Add entity client unit tests"

Table 6 The frequency and five-number summary of the expense amount for the most frequent eight software engineering tasks

Purpose (Stemmed word)	Freq.	Quantile value of corresponding expense amount					
		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Development (develop)	147	10	300	650	1,352	1,470	10,000
Bounty (bouti)	67	10	100	100	107	100	200
Maintenance (mainten)	48	50	178	500	1,654	1,400	9,000
Issue (issu)	33	25	360	480	507	720	1,230
Documentation (document)	25	50	398	975	771	1,063	1,318
Release (releas)	15	50	100	120	289	375	1,000
Testing (test)	12	50	88	140	351	618	1,180
Communication (communic)	0	0	0	0	0	0	0

5.3.2 Results

A total of 46% (665 out of 1,459) paid expenses are for engineering-related expenses. We filtered out 48% (318 out of 665) paid expense-related expenses, of which 132 of them contain multiple purposes and 186 of them do not contain at least one of the frequent words that we identified in Table 5. As a result of removing expenses that have multiple purposes and ones that do not contain any of the top 10 frequent words, 4 collectives are removed and 72 collectives remain. We ended up with 347 expenses (52%). According to the most frequent words in Table 5, we study the cost of the eight software engineering tasks that are mentioned in these 347 expenses.

40% (268 out of 665) engineering-related expenses involved development tasks (i.e., expenses that mention the word "develop") and the median cost of such expenses is $650. Table 5 shows the development tasks that are involved in most of the expenses (40%). Development tasks also co-occur with other software engineering tasks in the same expense. For the 268 engineering expenses that involved development, 121 of them mention other software engineering-related tasks (e.g., maintenance and documentation) as well. For example, the description of an expense says: "development and docs update in Nov". Table 6 shows that, in the expenses that are for development, the median expense amount is $650, which is the second-highest compared with that of the other tasks abovementioned.

18% (123 out of 665) of engineering expenses are due to a bounty or a specific issue (see Table 5). Both of the two stemmed words "bounti" and "issu" are related to the task of addressing issues. For the bounty expenses, collective maintainers first propose bounties on some issue reports of their GitHub projects, to motivate developers to address these issues (Zhou et al. 2020). After a developer addresses the issues, they submit an expense to claim the associated bounty. For issue expenses, developers first address the issue, then submit an expense to claim compensation for their effort.

The median cost for addressing one bounty issue is between $95 and $100, while for some specific issues, the cost can be as high as $930. There are 10 collectives with expenses that are related to bounties. Table 6 shows that the median cost of a bounty expense is $100 and the median cost for an issue expense is $465. Every bounty expense is proposed for addressing one issue, while an issue expense may represent the cost of addressing several issues. For example, the description of an issue expense says: "worked on issues #524, #549, #564, #558, #556". Hence, we manually extracted 85 issues by using the identified issue id (e.g., #54) from 24 (out of 34) issue expenses. Since there are no details on the cost of addressing each issue in an expense that was for addressing multiple issues, we estimated the average expense amount for each issue by the amount of the expense divided by the number of issue ids in the expense.

14% (97 out of 665) of the engineering expenses involve general maintenance tasks that only mention the keyword "maintenance" but do not mention a specific type of maintenance words (e.g., documentation), and the median cost of such expenses is $500. Table 5 shows that maintenance is the second most frequently mentioned word among engineering expenses. Table 6 shows that 48 engineering expenses are only for maintenance tasks and the median expense amount is $500. Documentation is a maintenance-related task. Table 6 shows that the median cost of documentation-related expenses is the highest at $975.

For the 132 removed expenses that have multiple purposes, a majority of them (90 %) have two purposes and 93 of them are related to development (i.e., containing the stemmed word "develop"). For the 26 expenses with two purposes that are not related to development, they are mainly related to maintenance and documentation

Non-engineering-related expenses represent 54.0% of all the expenses. "Web services", "marketing", and "travel" are the three most frequent and costly non-engineering-related expense types. For instance, 47% of marketing expenses are used for making stickers. For engineering-related expenses, the most frequent expenses are related to development and maintenance. Interestingly, we observe that 18% of the engineering expenses were spent to payout bounties for addressing issues with a median cost of $95.

6 RQ3: What are the Differences Between Individual-Supported Collectives and Corporate-Supported Collectives?

Motivation In Section 4, we observe different characteristics between individual and corporate donors. In this section, we investigate the differences between collectives that are mainly supported by individual donors (i.e., individual-supported collectives) and those that are mainly supported by corporate donors (i.e., corporate-supported collectives). For simplicity, we refer individual-supported and corporate-supported collectives as to *Ind_Collectives* and *Corp_Collectives*, respectively. For example, do Ind_Collectives and Corp_Collectives receive different donation amounts? Do they spend their funds differently? With a better understanding of the differences between Ind_Collectives and Corp_Collectives, the stakeholders of collectives could have a better expectation of their potential donors and expenses.

Approach We categorize Ind_Collectives and Corp_Collectives according to the proportion of the total received donation amount from individual donors and collective donors, respectively. For example, if we chose 20% as the threshold of the proportion, we consider the collectives in which more than 20% of their donation amount are from individual donors as Ind_Collectives and collectives in which more than 20% of their donation amount are from corporate donors as Corp_Collectives. Table 7 shows the descriptive statistics of collectives under different thresholds. We wish to study collectives that are mainly supported by individual donors or corporate donors. Hence, we chose 80% as the threshold empirically.

Table 7 The ratio between Ind_Collective and Corp_Collective using different thresholds

	Threshold					
	95%	90%	85%	80%	75%	70%
Ratio	7.6	3.7	2.2	1.9	1.7	1.5
(#Ind_Collective: #Corp_Collective)	(61:8)	(77:21)	(84:39)	(91:48)	(96:57)	(100:69)

We first study the differences between Ind_Collectives and Corp_Collectives in terms of the received total donation amount. Because different open source projects set up their collectives for receiving donations at different times and with different frequencies, we use the average received monthly donation amount (referred as to *monthly-donation-amount*) of a collective to represent its general received monthly donation amount. To determine whether the *monthly-donation-amount* between two types of collectives is statistically significant, we use the Wilcoxon rank-sum test and Cliff's delta d effect size.

Then we further compare these two groups of collectives in terms of the popularity of their associated projects on GitHub. For example, the more watches of an open source project has, the more users are interested in that project. We collected seven project-related metrics (namely, the number of issues, pull requests, watches, forks, contributors, stars, and commits) from GitHub to reflect the popularity of an open source project in GitHub. We also use the monthly bug reports number as a proxy of the project quality. To estimate the overview of the project quality, we calculate the average of the monthly bug reports number. We consider an issue as a bug if the issue report contains labels that have such bug-related keywords (i.e., "bug", "bugs", "defect", and "defects").

We also study the follow-up influence of donation on the community and maintenance activity of projects by taking monthly-based time-series data into consideration. We employed the next-month (i.e., a month after the month receiving donations) monthly community and maintenance-related metrics as proxies to measure the effects of donations. We used the number of new issues to represent the level of community activity of a project, and the numbers of closed issues and commits to represent the level of maintenance activity. Then we calculate the correlations between monthly donation amounts and the employed next-month monthly metrics to study the effects of donations on different groups of collectives.

Finally, we study the usage of different expense types for Ind_Collectives and Corp_Collectives. Similar to prior work (Hassan 2009), we employ Shannon's entropy to quantify the usage of expenses for each collective in terms of the expense amount across different expense types. The expense entropy of a collective quantifies the distribution of the expenses across the different expense types in a collective. A low entropy value for a collective indicates that the collective spent most of its funds on a small number of expense types. For example, if a collective's entropy is zero, the collective only spent funds on one specific expense type. A high expense entropy value for a collective indicates that the collective does not have a concentration for spending funds on specific expense types. For example, the expense entropy of a collective is one, indicating the collective spent funds on all occurrence expense types of the collective equally.

Results The monthly-donation-amount and total donation amount of Corp_Collectives are significantly higher than those of Ind_Collectives. Figure 18 shows the distribution of the monthly-donation-amount for Ind_Collectives and Corp_Collectives.

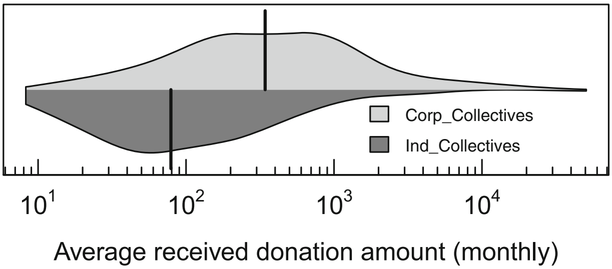


Fig. 18 The distribution of monthly-donation-amount for Ind_Collectives and Corp_Collectives

The median amount is \$343 for Corp_Collectives, while \$79 for Ind_Collectives. The Wilcoxon rank-sum test shows that there is a significant difference (p-value < 0.05) between them with a large effect size (Cliff's delta $= 0.50$), indicating the collectives driven by corporate donors received significantly more funds than the collectives driven by individual donors monthly. The median received donation amount of Ind_Collectives and Corp_Collectives are \$5,094 and \$1,406, respectively. The Wilcoxon rank-sum test shows that distributions of the total received donation amount are significantly different (p-value < 0.05) between Ind_Collectives and Corp_Collectives with a medium effect size (Cliff's delta $= 0.55$). Figure 19 shows the proportion of Ind_Collectives and Corp_Collectives under different ranges of total received donation amount. We observe that there are more Corp_Collectives than Ind_Collectives in the range that have larger received total donation amount. For example, when looking at the collectives with a total donation amount larger than \$10,000, 80% (24 out of 30) of them are Corp_Collectives. In other words, **Corp_Collectives are much more likely to get a large amount of donations (i.e., \$10,000 compared with Ind_Collectives**).

There is no significant difference between Ind_Collectives and Corp_Collectives in terms of the popularity and the quality of their associated GitHub projects. The Wilcoxon-rank test shows that there is no significant difference (p-values > 0.05) between Ind_Collectives and Corp_Collectives in terms of the number of issues, pull requests, watches, forks, contributors, stars, and commits of their associated project on GitHub.

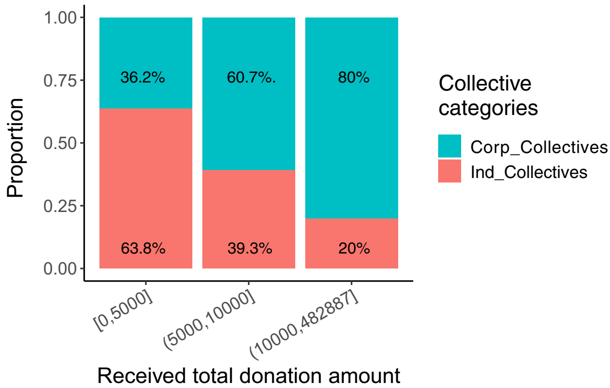


Fig. 19 The proportion of Ind_Collectives and Corp_Collectives under different ranges of total received donation amount

Table 8 The Spearman's rank correlation coefficient between next-month metrics and the monthly received donation amount of Corp_Collectives and Ind_Collectives

Next-month metrics	Monthly received donation amount	
	(Corp_Collectives)	(Ind_Collectives)
#New issues	0.46	0.35
#Closed issues	0.46	0.32
#Commits	0.36	0.35

In general, there is a positive correlation between the total donation of a project and its popularity. We compute the correlation between the total received donation amount and seven project-related metrics (i.e., the number of issues, pull requests, watches, forks, contributors, stars, and commits) using Spearman's rank correlation coefficient (Daniel et al. 1978). Among the calculated correlations (i.e., 0.435, 0.404, 0.404, 0.386, 0.370, 0.339, and 0.260, respectively), the number of issues has the highest correlation with the total received donation amount of an open source project.

The Wilcoxon-rank test also shows that there is no significant difference (p-value > 0.05) between Ind_Collectives and Corp_Collectives in terms of the average number of monthly bug reports amount of their associated project on GitHub. In terms of the average amounts of monthly received donations, both Corp_Collectives and Ind_Collectives have positive correlations (i.e., 0.45 and 0.31, respectively) with the average monthly bug reports amount. The possible explanation of the medium correlation in Corp_Collectives is that the corporate donors use the projects in their commercial product and in order to avoid the risk of using insufficiently maintained projects, they donate more to support the bug fixing.

We observe that **the monthly donation amounts are positively but weakly correlated with the community activity level and maintenance activity level in Ind_Collectives, while in Corp_Collectives, the correlations are stronger (see Table 8)**. One possible explanation for the stronger correlation between the monthly donation amount and the number of the next-month closed issues in Corp_Collective is that Corp_Collectives are more likely to get a larger amount of donations than Ind_Collectives (e.g., the median monthly received donation amounts are \$343 and \$79 in Corp_Collectives and Ind_Collectives, respectively).

Both Ind_Collectives and Corp_Collectives are likely to spend funds on a small group of specific types of expenses (e.g., engineering and web services). Figure 20 shows

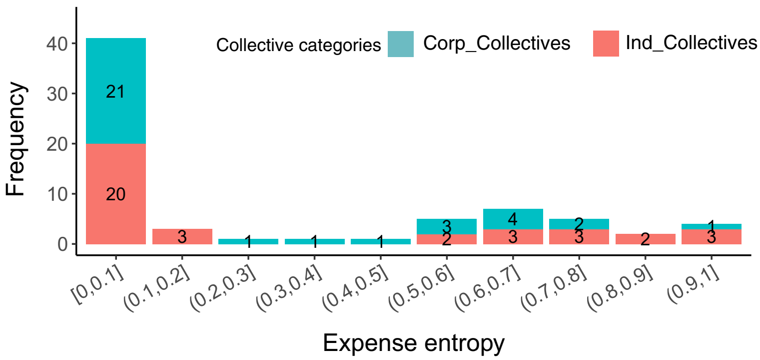


Fig. 20 The frequency for Ind_Collectives and Corp_Collectives' expense entropy

that the expenses of Ind_Collectives and Corp_Collectives have a similar pattern in terms of expense entropy. The entropy of 41 (59%) of the collectives is no more than 0.1, indicating those collectives spent funds on a very small group of specific expense types. In particular, 9 Ind_Collectives and 12 Corp_Collectives only spent funds on the engineering expense. Figure 21 shows the frequency of the most costly expense types in the 41 Ind_Collectives and Corp_Collectives with an entropy of no more than 0.1. We observe that the type of engineering is the most costly expense type in 45% (9 out of 20) Ind_Collectives and 71.4% (15 out of 21) Corp_Collectives. Except for the engineering expense, the web services expense is the main expense type for six of the Ind_Collectives.

Corp_Collectives have no significant difference in terms of the popularity of their associated GitHub projects. However, Corp_Collectives tend to receive a higher total and monthly donation amount than Ind_Collectives, and the monthly received donation amount has a stronger positive correlation with the level of community activity and maintenance activity in Corp_Collectives than in Ind_Collective. Both Ind_Collectives and Corp_Collectives are likely to spend funds on a small variety of expense types (e.g., engineering and web services).

7 Discussion

Given that our study is one of the first studies to explore the use of donations in an open source projects setting, we now discuss some peculiarities about donations and the use of donated funds in an open source setting.

In particular, we discuss some interesting findings in terms of rejected expenses, a successful case of bounty expenses, payment options of donors, the purposes of donation, and the other collectives that are not mainly supported by individual or corporate donors. Then we highlight the implications of our findings.

7.1 Rejected Expense Submissions

168 expense submissions were rejected in our data set. To analyze the rationale for such rejections, we first manually identified and filtered out 35 invalid rejected expense submissions which were done for testing the Open Collective platform's expense rejection

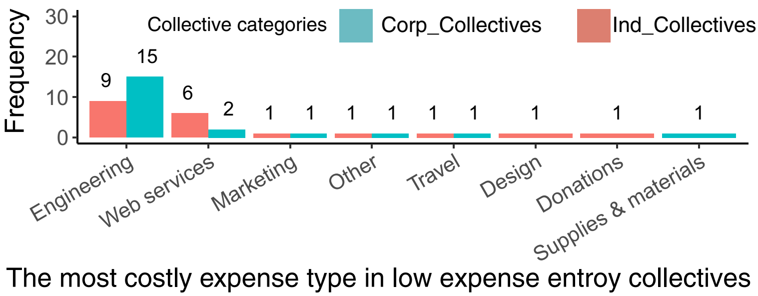


Fig. 21 The frequency of the most costly expense type in low expense entropy Ind_Collectives and Corp_Collectives

feature. Then we analyzed the 133 remained rejected expenses to understand which types of expenses are more likely to be rejected and whether there are users whose submission is always rejected. We first calculated the rejection rate for each expense type. Then we calculated the overall rejection rate for the users who submitted at least one rejected expense.

"Donation" expenses are more likely to be rejected. 42% of the "donation" expenses were rejected (i.e., a collective donating to another collective). In 13 collectives, 19 expenses were proposed as "Donation" type and eight of the 19 expenses were rejected.

We observe that several users performed suspicious activity, e.g., submitting several expenses with large amounts and vague reasons on the day that they just created their account. Before filtering out any rejected expenses, we identify 37 users whose expenses are always rejected. Some rejected expense amounts are large and the descriptions are meaningless. For example, a user *wassana-homchuen* submits three expenses to `Webpack` with the same value of $58,902 in one day with meaningless descriptions, i.e., "Available balance:", "http://www.90minlive.com", and "azuer". The user *japan-hunter* had similar behaviors to `Webpack`. Besides, both of these two user accounts were created on the date they submitted their expenses. We suspect that these two users want to "steal" donations from `Webpack`.

7.2 A Successful Case of Bounty Expenses

Bounty is a type of monetary incentive in open source projects. Users can propose bounties to motivate developers to complete tasks, which can be a bug-fixing task or a documentation task. In Section 5, we observe that bounty is a frequent purpose of expenses in engineering-related expenses. We observe that three collectives (i.e., the `Buttercup`, `Boostnote`, and `JHipster`) proposed bounties on issue reports. These bounties were paid out using the received donations of their corresponding collective. In total, there are 68 expenses related to a bounty with 91% of them being done in the `JHipster` collective. Especially, 77% (i.e., 62 out of 81) of the expenses in the `JHipster` collective were done to cover the costs of a bounty. The administrators of the `JHipster` collective explained that, with the growing user number, bounties were introduced to help manage the growing larger and more complex situation.[23] With a well-designed bounty rule[24] and the financial support from donations, 90.3% (i.e., 62 out of 67) bounty issue reports were addressed, which is much higher than the average addressing rate (i.e., 43.0%) of bounty issue reports (Zhou et al. 2020).

7.3 Payment Options for Donors

Donors can make donations using *Stripe*, *PayPal*, credit card, debit card, and gift card. **Stripe is the most popular payment processor for donors.** 97.2% donations are made through Stripe,[25] which is an online payment processors. **The gift card is a suggested payment method but still not a popular one.** Comparing with donating through credit cards or prepaid cards, the gift card is a more flexible payment method for corporations. By using gift cards, corporations can let their employees choose the collective that they might wish to support. Besides, we find that the median donation amount of gift cards is $20 which

[23] https://medium.com/open-collective/jhipsters-bounty-system-and-how-it-saved-the-project-cc118888f642

[24] https://www.jhipster.tech/bug-bounties/

[25] https://stripe.com/

Fig. 22 The word cloud of the top 10 frequent words from donation messages left by donors

is higher than the median donation amount (i.e., \$5) by other payment methods. Although the use of a gift card is officially recommended by the Open Collective platform,[26] only 1.2% of the donations were made using gift cards.

7.4 Purposes of Donation

43% of the donation messages express their gratitude to collectives. There are 41,471 donations after Oct. 6, 2017, of which only 589 of them had a donation message. Figure 22 visualizes the frequency of the top 10 frequent words in the donation messages. We observe that the top three frequent words are "thank", "work", and "great". 43% (i.e., 256 out of 589) of the messages contain either one of them. These three words are used to express gratitude to the collectives, e.g., "Thanks for doing great work". Especially we found six messages which express their appreciation to collectives for special release versions, for example, "GitExtension 3.0 release congrats". We also observe that the keyword "keep" in 11% (i.e., 64 out of 589) messages and this keyword expresses encouragements from donors, for example, "Keep going! Awooooo". Besides all these good words, we also observe a few cases which using the message for a successful donation contains an advertisement. For example, the contents of the six messages are about an online casino website.

7.5 The Collectives that are Not Mainly Supported by One Type of Donors

In Section 5, we study the Ind_Collectives (i.e., Individual-supported collectives) and Corp_Collectives (i.e., Corporate-supported collectives). We further study the remaining collectives, in which the donation proportion of individual donors are between 20% and 80%, by grouping them into three categories, namely Mainly-corporate-supported ($(20\%, 40\%]$), Other ($(40\%, 60\%]$), and Mainly-individual-supported collectives ($(60\%, 80\%]$). For simplicity, we refer to Mainly-corporate-supported collectives, Other collectives, and Mainly-individual-supported collectives as Mainly_Corp_Collectives, Other_Collectives, Mainly_Ind_Collectives, respectively. Table 9 shows the donation proportion of individual donors in all five collective categories.

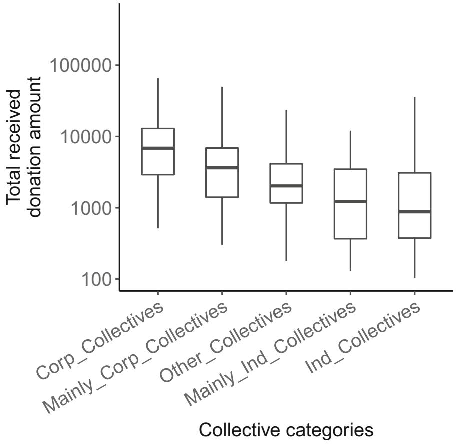
[26] https://docs.opencollective.com/help/backers-and-sponsors/gift-cards

Table 9 The donation proportion of individual donors in the five collective categories

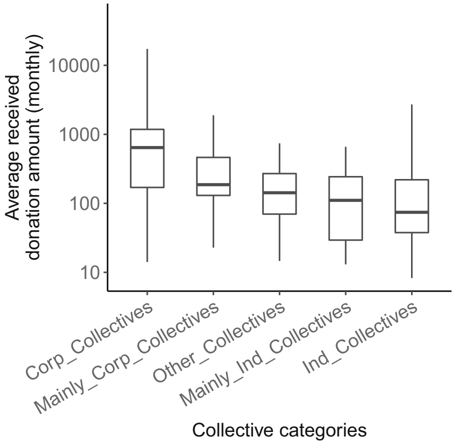
	Donation proportion of individual donors in a collective				
	(0%, 20%]	(20%, 40%]	(40%, 60%]	(60%, 80%]	(80%,100%)
Collective category	Corp_ Collectives	Mainly_Corp_ Collectives	Other_ Collectives	Mainly_Ind Collectives	Ind_ Collectives

Figure 23a and b show the distributions of the average received donation amount (monthly) and total received donation amount across different categories, including Corp_Collectives and Ind_Collectives. We observe that **in general, collectives with a higher proportion of donations from corporate donors are more likely to receive a larger amount of donations as expected**.

Similar to what we did for Ind_Collectives and Corp_Collectives in Section 6, we also employ Shanon's entropy to study the usage of different expense types for Mainly_Corp_Collectives, Other_Collectives, and Mainly_Ind_Collectives. Figures 24 and 25 show the frequency of expense entropy in the three categories of collectives, and the most costly expense type in low expense entropy collectives. We observe that all three categories of collectives tend to spend funds on a small group of specific types of expenses (e.g., engineering and web services) and this conclusion is coherent with what we observe from Ind_Collectives and Corp_Collectives. Since we also observe a similar expense usage style from Ind_Collectives and Corp_Collectives, we conclude that **all collectives tend to spend funds on a small group of specific types of expenses (e.g., engineering and web services)**.



(a)



(b)

Fig. 23 The the proportion of the donation amount between individual donors and corporate donors across collectives, under two selected metrics, (a) #commit and (b) #star, respectively

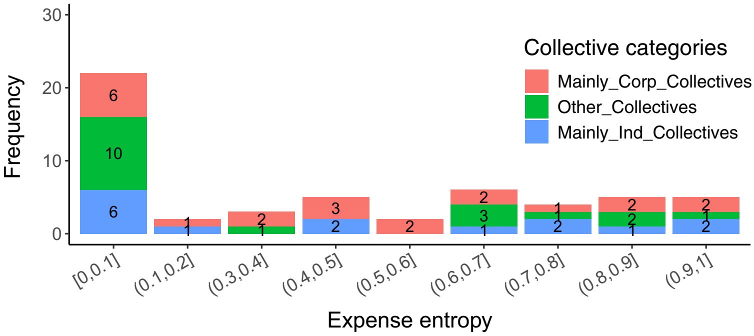


Fig. 24 The frequency for Mainly_Corp_Collectives, Other_Collectives, and Mainly_Ind_Collectives' expense entropy

7.6 Implications

Collectives should consider attracting more individual donors. In Section 4, we found that in general corporate donors donate more funds than individual donors in a single donation, but individual donates more money than corporate donors in total for a collective. One possible explanation is that individual donors tend to donate to a collective more consistently than corporate donors. This observation shows the importance of individual donors. Hence, the stakeholders of collectives should consider attracting more individual donors over corporate donors.

Collectives should not expect to receive a large amount of donations unless their associated projects are very popular and their projects are mainly supported by corporations. Figure 26 shows the trend of received donation amount of collectives against the number of issues of the in-associated open source projects. We observe that when the number of issues of a project reaches 9,000, the received donation amount is \$10,000. However, the likelihood of a GitHub project to reach such level of popularity is low. The mean and median number of issue requests of GitHub projects for different languages vary from 2.0 to 64.4 and 1 to 25, respectively (Bissyandé et al. 2013). Hence, collectives may not receive many donations from the community unless their projects are very popular. In addition, we observe that Corp_Collectives are much more likely to get a large amount of donations

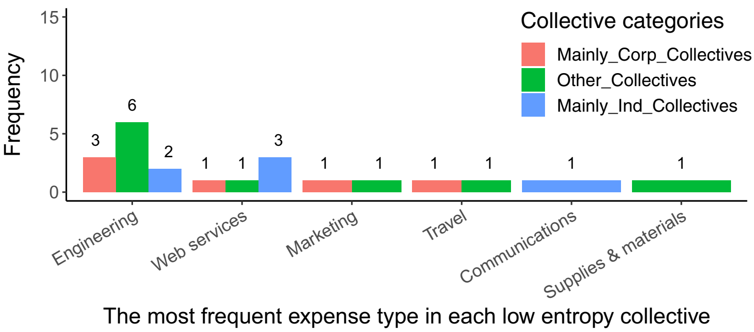


Fig. 25 The frequency of the most costly expense type in low expense entropy Mainly_Corp_Collectives, Other_Collectives, and Mainly_Ind_Collectives

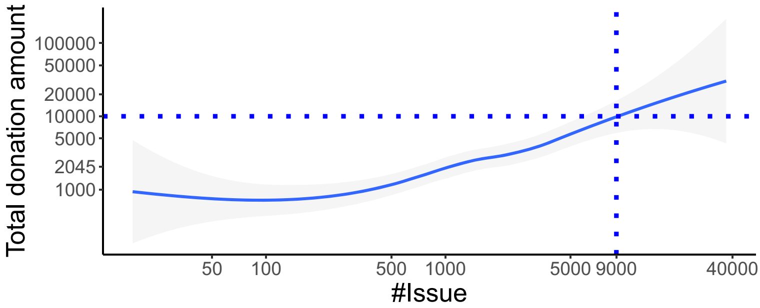


Fig. 26 The relationship between the total received donation amount of collectives and the number of issues of their associated GitHub projects. The donation amount is $10,000 when the number of issues reaches 9,000

(i.e., $10,000) compared with Ind_Collectives. Therefore, our findings suggested that collectives may not receive a large amount of funds from donations unless their projects are very popular and have corporations to support them.

Projects should budget for a reasonable amount (e.g., 13% of total funds) of non-engineering expenses when operating an open source project (e.g., marketing and travel). In Section 5.1, we observe 13% of the total expense amount is spent on non-engineering-related expenses. In Section 5.2, we show that 75.0% (104 of 139) of the collectives with expenses have non-engineering-related expenses and such expenses take up 45.6% (i.e., 665 out of 1,459) of their total expenses. In other words, non-engineering expenses are quite common in open source projects. For instance, two types of very frequent non-engineering expenses are marketing and travel. Therefore, we suggest that open source projects should allocate a reasonable amount of budgets for such non-engineering expenses.

Our study provides the overview of donations and expenses in open source projects and the characteristics of two specific types of collectives. We encourage future studies to investigate the associations between the characteristics of general collectives and donors, to provide insights on how to help collectives attract donors, such as which characteristics of collectives are more likely to attract donors to make donations. We also encourage future studies to investigate the associations between the evolution-related factors (e.g., the active user number of a project over time) of general collectives and expense-related factors, to provide insights on how to help collectives allocate their fundings.

8 Threats to Validity

Open source project donation is an area which is rapidly becoming a crucial area with the strong industrial support and involvement (GitHub 2019). This study is the first step in such an area. Threats to **external validity** are related to the generalizability of our findings. Among the mentioned donation platforms in Section 2, Open Collective platform is the only one that provides data for tracking expense transactions (at the time of our study). In this study, we focus on GitHub open source projects that are hosted on the Open Collective platform and our results may not be generalized to other code hosting platforms or donation platforms. Future research should study expenses from additional data sources to verify the generality of our findings on other platforms.

Threats to **internal validity** relate to the experimenter's bias and errors. One threat to internal validity is that we manually identified GitHub repositories of 102 studied collectives in Section 3, which may introduce bias due to human factors. Another internal validity is that we manually relabeled expenses types for expenses in Section 5 which may introduce bias. 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 (Gwet et al. 2002) to measure the inter-rater agreement. Before discussing differences, the Cohen's kappa coefficients are 0.94 for GitHub repositories identification and 0.91 for expense type relabeling, both indicating a high level of agreement.

One interval threat is that in Section 5.3, we estimate the cost of an expense with a specific purpose by using average amount of each issue (total amount of an expense/number of issues declared in the expense), which may include bias (e.g., addressing different issues in one expense may have different costs). We encourage future study to revisit the amount for each issue when data is available.

One internal threat to validity is that we empirically selected five as the threshold for the number of donors and \$100 as the threshold for the total donation amount in Section 3 to determine whether we should study a particular collective. Our threshold may introduce bias to our observations. Nevertheless, we have to select certain thresholds to filter out toy collectives and we believe the advantages of doing so outweigh the disadvantages.

One threat to the internal validity of our study is that we choose 80% as the threshold to tag a collective as being an individual or corporate supported collective. To alleviate this threat, we conducted a sensitivity analysis with a higher threshold of 90% and a lower threshold of 70%. Our findings still hold for both thresholds.

Another threat to the internal validity of our study is that we did not conduct surveys and interviews with developers. We made this decision due to the limitation of the public data that is available and ethical considerations. Money and bounties are a sensitive topic which are often framed in the context of larger discussions on fairness, stress, or even burnout, and unequal distribution of bounties on few participants or projects. Rather than adding stress to participants, we analyze the public artifacts, e.g., the expense descriptions left by developers.

9 Related Work

9.1 Online Donation-Based Crowdfunding

Crowdfunding is a specific form of crowdsourcing in business or finance, for raising funds for projects. Online donation-based crowdfunding is one of the practices of crowdfunding. Paredes et al. (2018) presented an overview of this practice. They developed a schema along with the donor, donation initiator, and platform/website dimensions to classify the donation-based crowdfunding platforms and further analyzed the characteristics of the studied platforms. Zhao et al. (2019) studied donor behaviours in donation-based crowdfunding. They proposed a deep survival model to analyze and predict the donation recurrence and donor retention in crowdfunding. They observed that donors with declared motives on their profiles are more likely to make more donations and stay for a longer period. Solomon et al. (2015) studied how the deadline (i.e., a donation timing strategy) affects the potential donors to decide when to donate projects. By simulating a crowdsourcing website, they found that in general, making early donations is a better strategy for donors since it will release a signal to others about the crowd's interest in the project and further attract more donations. In our study, we focus on the online donation-based crowdfunding that is for open source projects. In particular, we study the characteristics of two types of donors and we observe that individual donors are more likely to donate to the same collective recurrently than corporation donors.

9.2 Financial Incentives in Software Engineering

Financial incentives (e.g., donations and bounties) are being offered in open source projects to improve development and sustainability of open source projects. For example, open source projects collect donations to cover maintenance costs and maintainers propose bounties to attract and motivate developers to complete software development. Several studies have investigated how to improve the donations in software engineering. Krishnamurthy et al. (2014) studied why some open source developers accept financial rewards while others do not. They found that intrinsic (e.g., the need for a creative task) and extrinsic motivations (e.g., rewards) positively influence their willingness to accept monetary rewards, while community motivation (e.g., contributing to a social community) negatively influences their willingness to accept rewards. Krishnamurthy and Tripathi (2009) investigated the factors that impact community members' donation on open source platform (i.e., *Sourceforge*)[27] and found that the community members' time length of association with the community and rational commitment affect their donations. Yukizawa et al. (2019) investigated two psychological theories (i.e., social proof and legitimization of party contributions) to promote donations in open source projects. From an interview, they found that these two theories help promote and improve donations to open source projects. Nakasai et al. (2017) studied donations in the Eclipse project and found that the offered benefits to donors can motivate donations. Izquierdo and Cabot (2018) studied the role (e.g., an advisory role) of software foundations in open source projects. They analyzed the openness and the influence of 18 foundations in the development of open source projects. They observed most of the foundations' missions are providing legal support and leading evangelization actions.

The work of Overney et al. (2020) is closest to ours. They conducted a mixed-method empirical study to provide an overview of monetary donations in open source projects. They studied the prevalence of monetary donations in open source projects, the characteristics of projects, the common use of donations, and the effects of donations on project activities. The authors observed that a small fraction of GitHub open source projects have monetary donations, and that these projects are more active, more mature, and more popular than the ones without monetary donations. However, they did not find any strong evidence of the impact of monetary donations on the activity level of a project. Different from their work, we studied the characteristics of donors and their donations to help project operators better understand their donors and have a better view of the amount of donations. We also studied the use of donations but at a finer level of granularity (e.g., finer classifications of expense types and software engineering tasks). At the collective-level, we focused on two types of collectives that are supported by individual donors or corporate donors, respectively. We observe that popular and active collectives are more likely to receive donations from corporate donors. Additionally, we observe that the donation amounts are positively associated with the levels of community and maintenance activities in the collectives that are mainly supported by corporate donors.

[27] https://sourceforge.net/

Another group of studies focuses on studying the impact of bounties on software engineering tasks (e.g., bug-fixing). Nakasai et al. (2018) studied how donation badges (i.e., a widget used on the website to show a user has made donations) impact developers' responses to bug reports by a donor. They observed that Eclipse developers respond faster to the bug reports which are reported by users that have donation badges. Kanda et al. (2017) studied the issue reports that were offered bounties of GitHub projects and showed that the closing-time of bounty issue reports is longer than that of non-bounty issue reports. Zhou et al. (2020) performed a large scale study on more than 3,000 issue reports and found that the timing and bounty-usage frequency of a project are important factors in increasing the issue-addressing likelihood. Various studies analyzed the usage of bounties to motivate developers to detect and report software security vulnerability (Finifter et al. 2013; Maillart et al. 2017; Zhao et al. 2014). For example, Finifter et al. (2013) studied the vulnerability rewards program for Chrome and Firefox and found that compared to the cost of hiring a full-time security researcher, the vulnerability program is cheaper. Munaiah and Meneely (2016) studied the similarities and differences between two different vulnerability severity measurements: the vulnerability scores from the Common Vulnerability Scoring System (CVSS) and the monetary bounties from vulnerability reward programs (VRP). They observed a weak correlation between CVSS scores and bounty values, and they examined some potential reasons behind such differences. Maillart et al. (2017) suggested that project managers should dynamically adjust the value of rewards (e.g., money) according to the market situation (e.g., increase rewards when releasing a new version). Hata et al. (2017) conducted a user survey to understand the characteristics of vulnerability bounty hunters. They observed that most hunters are not project-specific and suggested VRP managers should strive to attract non-project-specfic security specialists with reasonable bounties. In our study, we observe several cases that applied donations to propose bounties to address issue reports.

Different from prior studies, we investigate the donors and their behaviour, and the usage of these donations on operating open source projects.

9.3 Understanding and Improving the Sustainability of Open Source Software Projects

Keeping open source projects sustainable is a challenging task. Therefore, researchers performed a significant number of studies on this topic to understand the sustainability of open source projects (Gamalielsson and Lundell 2014; Valiev et al. 2018; Coelho and Valente 2017; Eghbal 2016). Valiev et al. (2018) performed an empirical study to understand the relationship between the sustainability of a project and its surrounding projects (i.e., dependent projects or projects on which it depends) in the ecosystem. They showed that the number of project ties and the relative position in the dependency network has a significant impact on the sustainability of a project. Coelho and Valente (2017) investigated the reasons why modern open source projects fail and they found that failures are due to various reasons (e.g., lack of interest and time, low maintainability, and conflicts among developers). Eghbal (2016) reported the risks and challenges that are associated with maintaining open source projects, and argued that open source projects still lack a reliable and sustainable source of funds.

A number of prior studies studied the sustainability of open source projects from the angle of contributors (Lee et al. 2017; Ye and Kishida 2003; Avelino et al. 2016; Canfora et al. 2012; Pinto et al. 2016). For example, Avelino et al. (2016) found that 65% of their studied projects rely on one or two developers to survive. Lee et al. (2017) studied the motivations, and barriers that are experienced by the one-time code contributors. They found that the main motivation for one time contributors is to fix bugs that impeded their work. Such one-time contributors did not plan on becoming long term contributors due to various barriers, e.g., entry difficulties and lack of time. Ye and Kishida (2003) investigated the motivation of developers to participate in open source projects and found that the desire to learn is one of the major motivation and they also provided insights to improve the sustainability of open source projects based on their findings, e.g., creating a friendly environment and culture for newcomers to learn from the community. To help newcomers, Canfora et al. (2012) proposed an approach to identify and recommend mentors in software projects by mining data from mailing lists and version control systems. Steinmacher et al. (2016) proposed a portal, namely FLOSScoach, to support newcomers to open source projects.

In our study, we provide insights on how projects used their donated funds and find that development and web services expenses are the major expenses for open source projects.

10 Conclusion

In this paper, we studied 225 open source software projects that collected a total donation amount of $2,537,281 through the Open Collective platform. We first analyzed how donors make donations and how collectives use these donated funds. We found that:

1. In general, corporate donors tend to donate more funds than individual donors for an individual donation, while the total donation amount from individual donors are more than corporate donors in a collective, which suggests the influence of individual donors. Moreover, individual donors are more likely to redonate to a collective compared to corporate donors.
2. Non-engineering-related expenses take up 54.0% of the total number of all expenses. “Web services”, “marketing”, and “travel” are the three most frequent and costly non-engineering-related expense types. For engineering-related expenses, the most frequent expenses are related to development and maintenance.
3. Corp_Collectives are more likely to receive a larger total donation amount than Ind_Collectives collectives and the monthly received donation amounts are positively associated with the levels of community and maintenance activities in Corp_Collectives.

Our findings suggest that open source projects should not expect to receive a large amount of donations unless they are very popular. In general, open source projects should try to attract more individual donors since they tend to donate more and consistently in the long term than corporate donors. Our findings also instruct project operators about the need to budget for a reasonable amount of non-engineering expenses (e.g., marketing and travel).

Acknowledgements This research was partially supported by JSPS KAKENHI Grant Numbers JP18H03222.

References

Androutsellis-Theotokis S, Spinellis D, Kechagia M, Gousios G et al (2011) Open source software: a survey from 10,000 feet. Found Trends Technol Inf Oper Manag 4(3–4):187–347

Atiq A, Tripathi A (2016) Impact of financial benefits on open source software sustainability. In: International conference on information systems (ICIS), pp 1–10
Avelino G, Passos L, Hora A, Valente MT (2016) A novel approach for estimating truck factors. In: IEEE 24th international conference on program comprehension (ICPC), pp 1–10
Bauer DF (1972) Constructing confidence sets using rank statistics. J Am Stat Assoc 67(339):687–690
Bissyandé TF, Thung F, Lo D, Jiang L, Réveillère L (2013) Popularity, interoperability, and impact of programming languages in 100,000 open source projects. In: 2013 IEEE 37th annual computer software and applications conference. IEEE, pp 303–312
Canfora G, Di Penta M, Oliveto R, Panichella S (2012) Who is going to mentor newcomers in open source projects? In: Proceedings of the ACM SIGSOFT 20th international symposium on the foundations of software engineering (FSE), pp 1–11
Coelho J, Valente MT (2017) Why modern open source projects fail. In: Proceedings of the 2017 11th joint meeting on foundations of software engineering (ESEC/FSE), pp 186–196
Daniel WW et al (1978) Applied nonparametric statistics. Houghton Mifflin, Boston
Eghbal N (2016) Roads and bridges: the unseen labor behind our digital infrastructure. Ford Foundation
Finifter M, Akhawe D, Wagner D (2013) An empirical study of vulnerability rewards programs. In: USENIX Security symposium, pp 273–288
Gamalielsson J, Lundell B (2014) Sustainability of open source software communities beyond a fork: how and why has the libreoffice project evolved? J Syst Softw 89(1):128–145
GitHub (2019) GitHub Sponsors now supports projects! https://https://github.com/sponsors (last visited: Apr. 11)
Gwet K et al (2002) Inter-rater reliability: dependency on trait prevalence and marginal homogeneity. Stat Methods Inter-Rater Reliab Assess Ser 2(1):1–9
Hassan AE (2009) Predicting faults using the complexity of code changes. In: IEEE 31st international conference on software engineering (ICSE). IEEE, pp 78–88
Hata H, Guo M, Babar MA (2017) Understanding the heterogeneity of contributors in bug bounty programs. In: Proceedings of the 11th ACM/IEEE international symposium on empirical software engineering and measurement, ESEM '17, pp 223–228
Izquierdo JLC, Cabot J (2018) The role of foundations in open source projects. In: Proceedings of the 40th international conference on software engineering: software engineering in society (ICSE), pp 3–12
Kanda T, Guo M, Hata H, Matsumoto K (2017) Towards understanding an open-source bounty: Analysis of bountysource. In: IEEE 24th international conference on software analysis, evolution and reengineering (SANER), pp 577–578
Krishnamurthy S, Tripathi AK (2009) Monetary donations to an open source software platform. Res Policy 38(2):404–414
Krishnamurthy S, Ou S, Tripathi AK (2014) Acceptance of monetary rewards in open source software development. Res Policy 43(4):632–644
Lee A, Carver JC, Bosu A (2017) Understanding the impressions, motivations, and barriers of one time code contributors to floss projects: a survey. In: IEEE/ACM 39th international conference on software engineering (ICSE), pp 187–197
Long JD, Feng D, Cliff N (2003) Ordinal analysis of behavioral data. In: Handbook of psychology, pp 635–661
Maillart T, Zhao M, Grossklags J, Chuang J (2017) Given enough eyeballs, all bugs are shallow? Revisiting Eric Raymond with bug bounty programs. J Cybersecurity 3(2):81–90
Munaiah N, Meneely A (2016) Vulnerability severity scoring and bounties: why the disconnect? In: Proceedings of the 2nd international workshop on software analytics. SWAN 2016, pp 8–14
Nakasai K, Hata H, Onoue S, Matsumoto K (2017) Analysis of donations in the eclipse project. In: 8th International workshop on empirical software engineering in practice (IWESEP), pp 18–22
Nakasai K, Hata H, Matsumoto K (2018) Are donation badges appealing?: a case study of developer responses to eclipse bug reports. IEEE Softw 36(3):22–27
Overney C, Meinicke J, Kästner C, Vasilescu B (2020) 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 (ICSE), pp 1209–1221
Paredes H, Barroso J, Bigham JP (2018) All (of us) can help: inclusive crowdfunding research trends and future challenges. In: IEEE 22nd international conference on computer supported cooperative work in design (CSCWD). IEEE, pp 796–801
Pinto G, Steinmacher I, Gerosa MA (2016) More common than you think: an in-depth study of casual contributors. In: IEEE 23rd international conference on software analysis, evolution, and reengineering (SANER), vol 1, pp 112–123

Rich Sands (2012) Open source by the numbers. https://www.slideshare.net/blackducksoftware/open-source-by-the-numberss
Solomon J, Ma W, Wash R (2015) Don't wait! How timing affects coordination of crowdfunding donations. In: Proceedings of the 18th ACM conference on computer supported cooperative work & social computing (CSCW), pp 547–556
Steinmacher I, Conte TU, Treude C, Gerosa MA (2016) Overcoming open source project entry barriers with a portal for newcomers. In: IEEE/ACM 38th international conference on software engineering (ICSE), pp 273–284
Steinmacher I, Pinto G, Wiese IS, Gerosa MA (2018) Almost there: a study on quasi-contributors in open source software projects. In: Proceedings of the 40th international conference on software engineering, (ICSE), pp 256–266
Valiev M, Vasilescu B, Herbsleb J (2018) Ecosystem-level determinants of sustained activity in open-source projects: a case study of the pypi ecosystem. In: Proceedings of the 26th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering (ESEC/FSE), pp 644–655
Yamashita K, Kamei Y, McIntosh S, Hassan AE, Ubayashi N (2016) Magnet or sticky? Measuring project characteristics from the perspective of developer attraction and retention. J Inf Process 24(2):339–348
Ye Y, Kishida K (2003) Toward an understanding of the motivation open source software developers. In: Proceedings of the 25th international conference on software engineering (ICSE), pp 419–429
Yukizawa U, Tsunoda M, Tahir A (2019) Please help! a preliminary study on the effect of social proof and legitimization of paltry contributions in donations to OSS. In: 26th IEEE international conference on software analysis, evolution and reengineering (SANER), pp 609–613
Zhao M, Grossklags J, Chen K (2014) An exploratory study of white hat behaviors in a web vulnerability disclosure program. In: Proceedings of ACM workshop on security information workers (SIW), pp 51–58
Zhao H, Jin B, Liu Q, Ge Y, Chen E, Zhang X, Xu T (2019) Voice of charity: prospecting the donation recurrence & donor retention in crowdfunding. IEEE Transactions on Knowledge and Data Engineering (TKDE)
Zhou J, Wang S, Bezemer C-P, Zou Y, Hassan AE (2020) Studying the association between bountysource bounties and the issue-addressing likelihood of github issue reports. IEEE Transactions on Software Engineering (TSE)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Jiayuan Zhou is a senior researcher at Huawei Canada, Toronto Center for Software Excellence team. He received his Ph.D. from Queen's University, Canada. Before that, he spent two years working as a senior software developer in Alibaba Group. His research interests include mining software repositories, extrinsic incentives in crowdsourced software engineering, and intelligent vulnerability management. His work has been published in premier software engineering venues (e.g., ASE, TSE, and EMSE). More information at: http://jiayuan.dev.

Shaowei Wang is an assistant professor in the Department of Computer Science at University of Manitoba. He obtained his Ph.D. from Singapore Management University and his BSc from Zhejiang University. His research interests include software engineering, machine learning, data analytics for software engineering, automated debugging, and secure software development. His work has been published at premier venues like FSE, ASE, MSR and ICSM, as well as in major journals like TSE, TOSEM, and EMSE. He is one of four recipients of the 2018 distinguished reviewer award for the Springer EMSE (SE's highest impact journal). More information at: https://sites.google.com/view/mambalab/.

Yasutaka Kamei is an associate professor at Kyushu University in Japan. He has been a research fellow of the JSPS (PD) from July 2009 to March 2010. From April 2010 to March 2011, he was a postdoctoral fellow at Queen's University in Canada. He received his B.E. degree in Informatics from Kansai University, and the M.E. degree and Ph.D. degree in Information Science from Nara Institute of Science and Technology. His research interests include empirical software engineering, open source software engineering and Mining Software Repositories (MSR). His work has been published at premier venues like ICSE, FSE, ESEM, MSR and ICSM, as well as in major journals like TSE, EMSE, and IST. He served as a program-committee co-chair of the 23rd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER 2016) and the 15th International Conference on Mining Software Repositories (MSR 2018). More information about him is available online at http://posl.ait.kyushu-u.ac.jp/~kamei/.

Ahmed E. Hassan is an IEEE Fellow, an ACM SIGSOFT Influential Educator, an NSERC Steacie Fellow, the Canada Research Chair (CRC) in Software Analytics, and the NSERC/BlackBerry Software Engineering Chair at the School of Computing at Queen's University, Canada. His research interests include mining software repositories, empirical software engineering, load testing, and log mining. He received a PhD in Computer Science from the University of Waterloo. He spearheaded the creation of the Mining Software Repositories (MSR) conference and its research community. He also serves/d on the editorial boards of IEEE Transactions on Software Engineering, Springer Journal of Empirical Software Engineering, and PeerJ Computer Science. Contact ahmed@cs.queensu.ca. More information at: http://sail.cs.queensu.ca/.

Naoyasu Ubayashi is a professor at Kyushu University since 2010. He is leading the POSL (Principles Of Software engineering and programming Languages) research group at Kyushu University. Before joining Kyushu University, he worked for Toshiba Corporation and Kyushu Institute of Technology. He received his Ph.D. from the University of Tokyo. He was the general chair of AOSD 2013, ASPEC 2018, and MODELS 2021. He is a member of ACM SIGSOFT, IEEE Computer Society, and Information Processing Society of Japan (IPSJ). He is a Fellow of the IPSJ.

Affiliations

Jiayuan Zhou[1] · Shaowei Wang[2] · Yasutaka Kamei[3] · Ahmed E. Hassan[4] · Naoyasu Ubayashi[3]

Jiayuan Zhou
jiayuan.zhou1@huawei.com

Yasutaka Kamei
kamei@ait.kyushu-u.ac.jp

Ahmed E. Hassan
ahmed@cs.queensu.ca

Naoyasu Ubayashi
kamei@ait.kyushu-u.ac.jp

1 Centre for Software Excellence (CSE), Huawei, Canada

2 Department of Computer Science, University of Manitoba, Winnipeg, Canada

3 Principles of Software Engineering and Programming Languages (POSL), Kyushu University, Fukuoka, Japan

4 Software Analysis and Intelligence Lab (SAIL), Queen's University, Kingston, ON, Canada