An Analysis of Funding and Its Impact on Development Activities in Open Source Projects

Gakushi KIMURA†, Olivier NOURRY†, and Yoshiki HIGO†

† Graduate School of Information Science and Technology, The University of Osaka
1–5, Yamadaoka, Suita, Osaka, 565–0871, Japan

E-mail: †{g-kimura,nourry,higo}@ist.osaka-u.ac.jp

Abstract Financial support has become important for sustaining Open Source Software (OSS), but how funds are spent and how they relate to development activity remain unclear. This study analyzes OSS projects in Open Collective by combining transaction histories with GitHub commit histories. We found that funding consists of many small contributions and fewer, larger expenses, and that development-related spending is the largest identifiable expense category. Individual payments are mainly directed toward existing contributors, and projects with substantial development-related spending show more favorable commit growth. These findings suggest that OSS funding should be analyzed by focusing not only on funding amounts, but also on spending purposes and recipients.

Key words Open Source Software, Sustainability, Funding, Development Activity, Empirical Study

1. Introduction

Open Source Software (OSS) has become an essential foundation of modern software systems. Many companies, public organizations, researchers, and individual developers rely on OSS libraries, frameworks, and tools in their daily activities [1]. However, sustaining OSS projects is difficult because many projects depend on voluntary contributions from a limited number of developers [2]. When these developers lack sufficient time, resources, or incentives to continue maintenance, projects may become inactive, even if many users depend on them [3].

To address this sustainability problem, OSS projects increasingly use funding platforms such as Open Collective(1) and GitHub Sponsors(2). These platforms allow projects to receive financial support from individuals, companies, and other organizations. Such funding may help projects cover infrastructure costs, hire contributors, organize events, or support maintenance and development work. Therefore, funding platforms are becoming an important mechanism for supporting the long-term sustainability of OSS projects.

As OSS projects are increasingly joining these funding platforms (and possibly depending on them) to receive financial support, we have yet to understand the relationship between financial contributions and sustained development activities. To understand the role of funding in OSS sustainability, it is necessary to examine how much money flows into OSS projects, how that money is spent, who receives it, and whether development-related spending is associated with development activity. Existing discussions often emphasize the importance of funding for OSS, but the actual financial flows and their relationship with development activity remains unclear [4].

Open Collective provides a useful opportunity to investigate this problem due to it transparent transaction histories being publicly visible. Unlike many other funding mechanisms, Open Collective records detailed financial transactions, including contribution amounts, expense amounts, transaction dates, descriptions, and account types. By linking these financial records with GitHub repositories, we can analyze both funding activity and software development activity for the same OSS projects.

In this study, we analyze OSS projects on Open Collective to understand how funding is received, how it is spent, and how development-related spending is associated with commit activity. Specifically, we address the following three research questions:

(1) **RQ1: What are the overall characteristics of funding in OSS projects on Open Collective?**
(2) **RQ2: How does money flow among different stakeholders in Open Collective?**
(3) **RQ3: Is development-related spending associated with changes in commit activity?**

To answer these questions, we collected Open Collective transaction records and GitHub commit histories of participating OSS projects. We first characterize the overall scale, transaction types, temporal trends, and expense purposes of OSS funding on Open Collective. We then analyze money flows among stakeholder account types and examine whether payments are made to compensate existing contributors or to hire new contributors. Finally, we investigate whether projects with higher development-related spending show different commit activity changes after joining Open Collective.

The contributions of this study are as follows. First, we provide an empirical characterization of OSS funding on Open Collective using large-scale transaction data. Second, we analyze how money flows among different stakeholders and identify the role of individual recipients in OSS funding. Third, we examine the relationship between development-related spending and commit activity.

2. Background and Related Work

2.1 OSS Sustainability and Funding

OSS has become a fundamental component of modern software systems. However, many OSS projects depend on voluntary work by

(1): https://opencollective.com/
(2): https://github.com/open-source/sponsors

few of developers, making long-term maintenance difficult. Serious incidents such as Heartbleed(3) in OpenSSL [5] and Log4Shell(4) in Log4j have highlighted the risks of under-resourced maintenance in widely used OSS. These cases show that OSS sustainability is not only a technical issue, but also a resource allocation problem.

Financial support through donations, sponsorships, grants, and funding platforms has therefore become important for sustaining OSS projects. It can help projects hire contributors, pay for infrastructure, and support maintenance activities. However, prior work has not sufficiently clarified how funds are actually spent, who receives them, and how such spending relates to development activity.

2.2 Open Collective

Open Collective is a funding platform that allows OSS projects and communities to receive and manage funds. A distinctive feature of Open Collective is financial transparency: contributions and expenses are publicly visible as transaction records. This makes Open Collective a useful data source for studying OSS funding as financial flows rather than only as aggregate donation amounts.

Compared with platforms that mainly expose sponsor counts or total funding amounts, Open Collective enables more detailed analysis of how OSS projects receive and spend money.

2.3 Development Activity Metrics

Empirical studies of OSS projects often use repository-level metrics such as commits, pull requests, issues, and contributors to measure development activity [6]. Among these metrics, commit count is a scalable proxy for development activity because it can be consistently collected across GitHub and compared over time.

Still, commit count does not capture all forms of OSS maintenance, such as code review, issue triage, release management, documentation, or community support. Therefore, in this study, commit activity is interpreted as one observable indicator of development activity, rather than a complete measure of OSS maintenance.

3. Methodology

This section describes the dataset construction, account matching, and expense classification procedures used throughout this study. The detailed analysis procedure for each research question is described in Section 4.

3.1 Open Collective and GitHub Data Collection

We collected two types of data: financial information from Open Collective and historical development activity from GitHub. In Open Collective, a unit that receives and manages funds is represented as a *collective*. In this paper, we use the term *collective* to refer to an Open Collective account that manages funds, and the term *project* to refer to the corresponding OSS project, typically represented by a GitHub repository.

For each Open Collective record, we extracted transaction information including transaction dates, amounts, currencies, transaction kinds, descriptions, and account types. Most transaction records represent money received or spent by collectives, while a subset of records represents money received or spent directly by projects.

To analyze the relationship between funding and development activity, we needed to connect Open Collective projects with their corresponding GitHub repositories. We therefore identified GitHub repositories using project metadata available on Open Collective, such as project URLs and `github_handle` fields. We then mined each repository's commit history from GitHub. This linkage enabled us to link and analyze financial transactions and development activity for a given OSS project.

(3): https://www.heartbleed.com/

(4): https://www.ipa.go.jp/archive/security/security-alert/2021/alert20211213.html

Table 1 Summary of the Dataset

Item	Count
Collectives	2,624
Projects	522
Transactions	2,389,362
Cloned GitHub repositories	1,378

Table 1 summarizes the dataset. We find 2,624 collectives, 522 participating OSS projects, and over 2.3 million transactions. We were also able to find 1,378 GitHub repositories associated with either a collective or a project. The large number of transaction records indicates that Open Collective is both very active and important for OSS projects, and allows us to analyze OSS funding as a sequence of financial flows rather than as a single aggregate donation amount.

3.2 Transaction Data Processing

Open Collective transactions are recorded in multiple currencies. To make transaction amounts comparable across currencies, we converted all monetary values into USD. The exchange rates used in this study were those available at the time of the experiment, in June 2026. We also standardized the direction of financial flows using the *transaction kind* field provided by Open Collective. In this study, `CONTRIBUTION` records represent money entering Open Collective entities, while `EXPENSE` records represent money leaving collectives or projects. This distinction allows us to separately analyze incoming funding and outgoing spending for our analyses.

3.3 Matching Fund Recipients to GitHub Contributors

To examine whether individual recipients were actual project contributors, we matched Open Collective recipients with GitHub contributors. This was necessary because Open Collective records alone do not indicate whether a payment recipient had contributed to the corresponding repository.

We focused on transactions whose recipient account type was `INDIVIDUAL`. For such recipients, we compared the registered name on Open Collective with contributor names in the corresponding GitHub repository. To reduce superficial mismatches, we normalized names by ignoring differences in spaces and hyphens. We then treated a recipient as matched when the normalized Open Collective name matched a normalized GitHub contributor name.

This matching procedure allowed us to examine whether individual recipients had contributed commits to the corresponding repository before receiving payment. In other words, it enabled us to distinguish between payments to existing contributors and payments to individuals who started contributing only after receiving funding.

3.4 Designing Transaction Categories and Classifying Expense Purposes

Open Collective transaction records include a textual `description` field, which provides clues about the purpose of each financial transaction. However, not all descriptions clearly state the purpose of the expense. Some records contain only short or ambiguous descriptions, and some require additional context that is not available in the transaction data. Therefore, we designed expense-purpose categories and refined the LLM-based classification prompt through an iterative process.

The procedure was as follows:

(1) We first manually inspected the mined Open Collective transaction data to understand what kinds of expenses appeared in the dataset, and created an initial list of expense-purpose categories.
(2) Using the initial categories, two authors then independently labeled a statistically significant random sample of 381 expense records, drawn from the full set of 50,458 records, corresponding to a 95% confidence level with a ± 5% margin of error.
(3) The two authors then compared their labels. For records where

Table 2 Definitions of Expense Categories

Category	Description
Development	Compensation paid to official project members for direct software development and maintenance.
Infra & Subscription	Recurring costs for cloud hosting, internet connectivity, and other software-as-a-service subscriptions.
Equipment	Purchase of physical hardware and assets directly used for development activities (e.g., laptops and servers).
Food & Supplies	Purchase of consumables, meals, and general physical items that are not directly related to development.
Marketing & Events	Costs for organizing or participating in events to promote the project and recruit new developers, including promotion, transportation, and registration fees.
Non-tech Activities	Essential tasks not directly linked to software development, translation, legal or tax compliance, and other miscellaneous fees.
Unknown	Expenditures where the purpose cannot be determined due to missing or insufficient information.

the labels differed, they discussed the case and created an agreed-upon label. We used these agreed-upon labels as reference labels for the next iteration until no new labels emerged. The Cohen Kappa score [7] for the manual labeling was 0.65, indicating a substantial agreement between the two authors.

(4) We provided GPT-5.4 with the transaction `description` and the definition of each expense-purpose category derived from the manual labeling. For each transaction in the subset, we asked it to output a category label, a confidence score, and the reason for selecting that label. This process was repeated 5 times for each sample with a temperature set to 0 to reduce variance.

(5) We compared the LLM-generated labels with the reference manual labels. When the LLM classification differed from the reference label, we examined the reason generated by the LLM and used it to identify why the prompt led to an incorrect classification. We then refined the prompt so that the LLM classifications better matched the human judgments.

(6) Finally, we sampled another distinct set of 381 expense records and repeated the manual labeling and prompt refinement process to evaluate the model's performance.

This process resulted in seven categories, as shown in Table 2. We included the **Unknown** category to avoid forcing ambiguous expenses into inappropriate categories when the transaction description did not provide enough information to determine the purpose. The overall accuracy of the LLM classification was 81.8% for the sampled records. When we restricted the evaluation to records with confidence scores of at least 0.90, the accuracy increased to 96.6%. Therefore, we decided to only use records whose classification confidence was at least 0.90.

To obtain a statistically relevant sample of 1,784 samples (99% ± 3%) with a confidence of 0.90 or more, we iteratively labeled random subsets of 1,784 transactions using GPT-5.4. After three iterations, we ended up with 2,163 records with a confidence of 0.90 or above. The result of this multi-class classification was used to summarize the broader distribution of expense purposes as shown in Section 4.

Table 3 Transaction Records by Kind

Kind	Count	Amount (USD)
CONTRIBUTION	1,080,778	42.50M
HOST_FEE	825,224	-5.15M
PROCESSOR_FEE	413,958	-0.66M
EXPENSE	50,458	-74.32M
ADDED_FUNDS	14,522	65.26M
Others	4,422	-0.68M

3.5 Large-scale Labeling of Development-Related Transactions

We performed one additional binary classification to identify development-related spending for project-level analyses. This classification was applied to all transaction records whose `kind` was `EXPENSE`. To reduce false positives, we adopted a conservative rule. We ran the binary classification five times by using GPT-5.4 mini for each expense record and treated an expense as development-related only when all five runs labeled it as development-related with confidence scores of at least 0.90 (a confidence score of 0.90 matched 98% of the manually labeled sample). This rule was intended to identify expenses that were consistently and confidently classified as development-related. The results shown in Section 4.2 therefore include only samples with a confidence score above 0.90.

4. Results

4.1 RQ1: What are the overall characteristics of funding in OSS projects on Open Collective?

Motivation. Before discussing whether funding supports development activity, we need to understand the basic financial landscape, how many flows through Open Collective, what kinds of transactions are common, and how OSS projects spend their funds. In RQ1, we therefore characterizes the overall scale, transaction types, temporal trends, and expense purposes of OSS funding on Open Collective.

Approach. We analyzed the Open Collective dataset from three perspectives: transaction scale, temporal trends, and expense purposes. First, we summarized the number of collectives, projects, transaction records, and linked GitHub repositories. We then aggregated transaction records by `kind` and calculated both transaction counts and total amounts in USD. Using median and mean values, we then compare incoming (`CONTRIBUTION`) and outgoing (`EXPENSE`) transaction data. Next, we examined yearly OSS funding trends by aggregating `CONTRIBUTION` amounts by year. The value for 2025 was treated as a partial-year observation because our data mining period ended in October 2025.

Finally, using the expense-purpose labels described in Section 3, we analyzed how projects spend their funds. We focused on records classified with confidence scores of at least 0.90 (2,163 records) and summarized the distribution of expense categories by both transaction count and monetary amount.

Results. Table 3 shows transaction records by kind. `CONTRIBUTION` represents incoming contributions, `EXPENSE` represents outgoing payments, `HOST_FEE` and `PROCESSOR_FEE` represent fiscal-host and payment-processing fees, and `ADDED_FUNDS` represents funds added outside the platform. `CONTRIBUTION` and fee-related records dominate the dataset, while `EXPENSE` records are less frequent but larger in total monetary volume. This suggests that Open Collective funding is characterized by frequent incoming contributions and less frequent but larger outgoing payments.

Table 4 further illustrates this contrast. Contributions were typically small, with a median amount of USD 5.00, whereas expenses were substantially larger, with a median amount of USD 200.00. The mean values are much larger than the medians for both transaction

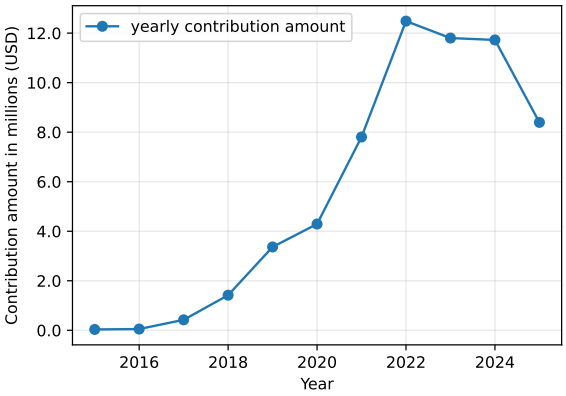
Table 4 Transaction Statistics for Contributions and Expenses

Metric	Contribution	Expense
Count	1,080,776	50,457
Total (USD)	61.85M	79.11M
Mean (USD)	57.22	1,567.80
Median (USD)	5.00	200.00

types, indicating that both contribution and expense amounts are highly skewed. These results suggest that Open Collective aggregates many small incoming payments and redistributes them through fewer, larger expenditures.

Figure 1 shows the yearly trend in contribution amount and how OSS funding evolved through Open Collective changed over time. Increasingly larger sums of money are being funneled into open-source projects over time. Note that the the contribution amount for 2025 represents only a partial-year (until October) and the total shown likely smaller than the actual annual amount.

Fig. 1 Yearly trend of contribution amounts to OSS projects on Open Collective.



Finally, we summarize the expense-purpose classification results described in Section 3 for the 2,163 records with 0.90 or higher classification confidence. As shown in Table 5, development-related expenses were the largest category in both transaction count and monetary amount, suggesting that direct development and maintenance work is a major identifiable use of OSS funds on Open Collective. Infrastructure and subscription costs were also frequent, indicating that projects also spend funds on operational resources required to maintain development environments and services.

Table 5 Expense Purpose Distribution in the Sampled Records

Category	Count	Count (%)	Amount (%)
Development	920	42.5	44.5
Infra & Subscription	630	29.1	2.7
Marketing & Events	221	10.2	3.8
Non-tech Activities	131	6.1	8.3
Unknown	92	4.3	38.6
Equipment	85	3.9	1.2
Food & Supplies	84	3.9	0.9

While the unknown category accounts for a small share of records, we find that it represents a large share of the total amount. Indeed, our results show that a small number of high-value expenses could not be reliably classified from the available transaction information. Therefore, development-related spending should be understood as the largest identifiable category, rather than as a complete description of all high-value expenses. These results highlight potential shortcomings of the funding platform as significant amount of donation money can be spent without being required to provide a description or details about the purpose and use of the funds.

> We find that donations toward OSS projects are steadily increasing over time and that development-related spending is the largest identifiable expense category.

4.2 RQ2: How does money flow among different stakeholders in Open Collective?

Motivation. While RQ1 showed that Open Collective aggregates many small contributions and redistributes them as fewer, larger expenses, it provides no insight into how money flows within the OSS ecosystem. This distinction is important because OSS funding is often discussed as a way to support maintainers and developers, but funds may also be transferred to organizations, vendors, other projects, or collectives.

Therefore, we analyze money flows among account types to understand whether funds are mainly directed to individual contributors or between larger organizations. For payments made to individual, we further wanted to investigate if projects are compensating existing contributors, or if funds are used to pay new developers. If most individual payments go to people who had already contributed before receiving money, funding may primarily function as support or compensation for existing contributors. In contrast, if many recipients start contributing only after receiving money, funding may also function as a mechanism for attracting new developers, highlighting the challenge of attracting open-source developers. We first analyzed money flows among account types.

Approach. Using the `from_account_type` and `to_account_type` fields provided by Open Collective, we analyzed the flow of money between stakeholders throughout the OSS ecosystem. For both `CONTRIBUTION` and `EXPENSE` transactions, we aggregated the total amount in USD for each pair of sender and recipient account types. We then visualized these amount-based flows as heatmaps.

Using the binary classification results described in Section 3, we then focused on individual recipients of development-related payments. Specifically, we identified `EXPENSE` records whose recipient account type was `INDIVIDUAL` and whose purpose was classified as development-related in all five LLM classification runs with confidence scores of at least 0.90. This conservative criterion was used to focus on payments that were consistently identified as direct development-related payments.

Because Open Collective and GitHub use different account systems, we matched individual recipients to GitHub contributors using account names. This matching process identified 456 individual recipients whose Open Collective account names matched GitHub account names in the corresponding repositories. For each matched individual recipient, we investigated their commit activity starting from one year before, to one year after the recipient started getting remunerated using OSS donations. This analysis allowed us to examine whether individual recipients were already contributors before receiving money (suggesting the project wants to ensure it does not lose its developers), or started contributing after receiving money (suggesting they were hired by the OSS project).

Results. Figure 2 shows the money flow among account types for `CONTRIBUTION` transactions. The heatmap shows that contributions are mainly directed toward `COLLECTIVE` accounts. In particular, large flows from `ORGANIZATION` and `INDIVIDUAL` accounts to `COLLECTIVE` accounts indicate that Open Collective receives funding from both organizational and individual contributors. We also observe non-negligible flows from `FUND` and `COLLECTIVE` accounts

to PROJECT accounts, suggesting that some funding is redistributed within the Open Collective ecosystem before reaching project-level accounts and individuals.

Fig. 2 Money flow of CONTRIBUTION transactions by account type.

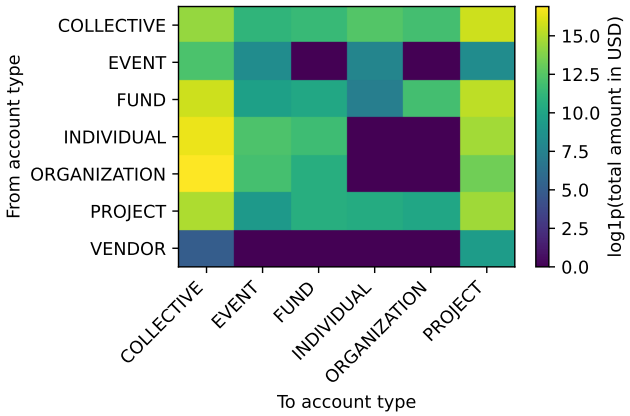
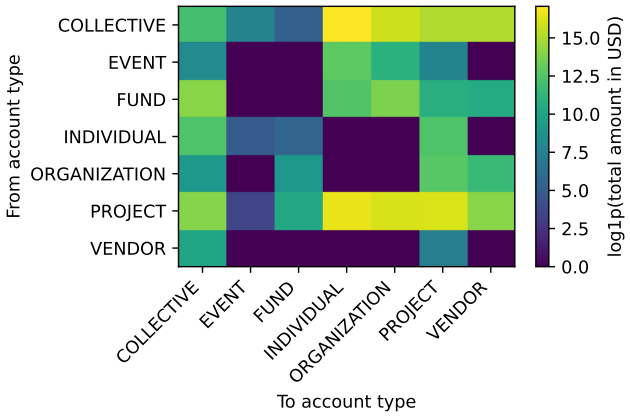


Figure 3 shows the corresponding money flow for EXPENSE transactions. In contrast to contributions, expenses are largely directed toward INDIVIDUAL accounts. As shown in the figure, the largest flows are from COLLECTIVE or PROJECT accounts to INDIVIDUAL accounts. However, we also find a non-trivial amount of transactions between ORGANIZATION, PROJECT, and VENDOR accounts. These results indicate that while a substantial portion of outgoing funds is paid to individual recipients, a non-trivial amount of OSS project funds are also used for organizational payments, inter-project transfers, and service-related expenses.

Fig. 3 Money flow of EXPENSE transactions by account type.



We then examined a one year window of commit activity for individual recipients who provided development-related services to determine if they were already contributing to the projects from which they received funds, and if existing contributors' development activities were impacted from receiving payment.

Table 6 Commit Activity of Matched Individual Recipients Before and After First Development Payment

Status	Percentage
Prior contributor, increased commits	30.9
Prior contributor, decreased commits	56.6
New contributor after payment	3.5
No commits before or after payment	9.0

As shown in Table 6, most individual recipients had already committed to the corresponding repository before they started receiving compensation for development activities. This suggests that open-source funding are commonly used to retain existing developers rather than attract new ones. Among prior contributors, more recipients exhibited lower commit activity than increased commit activity after receiving payment, indicating that getting compensated does not automatically translate with increased coding activity. A small share of recipients started committing only after receiving payment, suggesting that funding may also be associated with new contributor participation in some cases.

> We find that contributions tend to flow from organizations and individuals to collectives, while expenses flow toward individual recipients. Development-related payments primarily go to existing contributors.

4.3 RQ3: Is development-related spending associated with changes in commit activity?

Motivation. RQ1 showed that development-related spending is the largest identifiable expense category on Open Collective. RQ2 further showed that many outgoing expenses are directed toward individual already involved to the OSS projects on GitHub. However, these results do not show whether development-related spending impacts development activity.

This question is important because OSS funding is often expected to support the sustainability of development and maintenance work. Once a project joins Open Collective and starts receiving and spending donation money for software development, we expect projects to maintain or increase development activity. Moreover, we hypothesize that projects that spend a larger percentage of their funds on development see a bigger impact on development activity than projects that tend to spend on non-development related activities. Therefore, RQ3 examines the relationship between the proportion of spending allocated to development and changes in commit activity.

Approach. To answer RQ3, we measured development activity using GitHub commit counts. For each project, we used the Open Collective registration date as the reference point and counted monthly commits from 12 months before to 12 months after registration. We first analyzed the overall trend in mean and median monthly commits to examine whether joining Open Collective coincides with an increase in development activity.

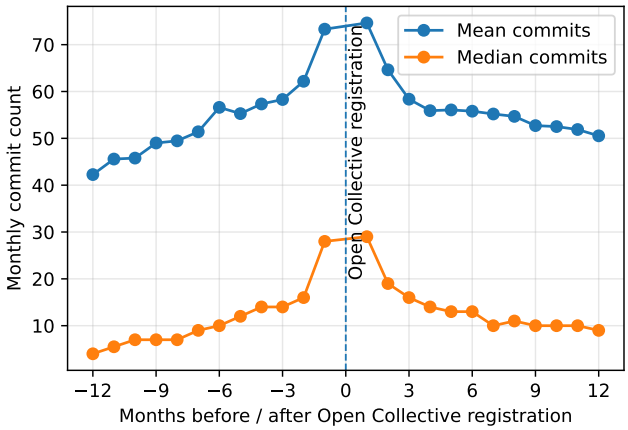
We then analyzed whether development-related spending is associated with commit activity changes. For each project, we calculated the proportion of expense spent toward development by dividing the development-related expense amount by the total expense amount. Based on this ratio, we divided projects into three tertiles: bottom 33%, middle 33%, and top 33%. For each group, we calculated the commit growth rate over the 24-month window before and after Open Collective registration.

Finally, we compared commit growth rates among the three groups. We used the Kruskal–Wallis test to examine whether there was a statistically significant difference among the groups. We then performed pairwise Mann–Whitney U tests with Bonferroni correction to identify which group pairs differed. This analysis does not aim to establish a causal effect of development-related spending. Instead, it examines whether development-related spending is empirically associated with commit activity changes.

Results. Figure 4 shows the mean and median number of monthly commits from 12 months before to 12 months after joining Open Collective across projects. We find that the median number of commits tends to steadily increase over time across projects, peaking when joining Open Collective, then sharply decrease afterward with no signs of recovery. Using a Wilcoxon test, we find that the number

of commits prior and after joining Open Collective is statistically significant ($p = 0.006$), indicating that the decrease in commit is not random. The mean number of commits followed a similar pattern, though larger than the median due to highly active projects.

Fig. 4 Monthly commit activity around Open Collective registration.



These results show that projects tend to join Open Collective while they are very active, possibly looking to obtain funding to increase their development activities. However, our results suggest that registration itself does not necessarily coincide with increased or sustained commit activity. Therefore, rather than considering the availability of funding alone, we further analyze how the amount of funds spent towards development can impact development activity.

Table 7 shows the commit growth rates based on the top, middle, and lowest spender of funds on development activities. Interestingly we find that a significant portion of OSS projects on Open Collective are choosing to not spend any donation funds on development and maintenance (as shown by the mean and median values of the bottom/middle groups). These results therefore show that projects that do not spend funds on development (as shown by the bottom and middle groups) tend to display negative median growth rate over time, whereas projects that spend significant funds on development (i.e., the top group) tend to maintain or increase development activities.

Table 7 Commit growth by development-related spending group. Dev. exp. denotes development-related expense in USD.

Group	N	Growth	Mean dev. exp.	Median dev. exp.
Bottom 33%	214	-18.2%	0.00	0.00
Middle 33%	214	-3.6%	82.39	0.00
Top 33%	213	6.1%	18,983.32	3,275.00

We conducted statistical tests to examine whether the difference in commit growth among the three groups were significant. A Kruskal–Wallis test showed a statistically significant difference among the groups ($H = 6.20$, $p = 0.045$). We then performed pairwise Mann–Whitney U tests with Bonferroni correction and found a statistically significant difference between the bottom 33% group and the top 33% group.

We find that a large proportion of projects do not spend donations towards development and maintenance. Our results also show that projects that spend a larger proportion of donations on development exhibited higher development activity.

5. Discussion

Our findings indicate that OSS funding should be examined not only by the total amount received, but also as a flow of resources among contributors and organizations. Open Collective aggregates many small contributions and redistributes them through fewer, larger expenses, suggesting that funding platforms act as financial coordination mechanisms for OSS projects.

From RQ1 and RQ2, we find that development-related spending is the primary expense category, both in terms of frequency and total amount. We also find that in most cases, development expenses are paid to existing contributors. This suggests that funding may often serve as a mean to maintain or continue development, rather than functioning as a mechanism to recruit new developers.

From RQ3, we find that development activity commonly decreases after projects join Open Collective. Upon closer inspection, we find that a significant portion of OSS projects do not allocate funding to development. Projects that do spend on development are more likely to maintain or increase development activity, whereas those that do not tend to experience a significant decrease. These results suggest that donations can support OSS development activity when funds are explicitly allocated toward software development.

6. Conclusion

This study investigated how OSS funding relates to development activity by analyzing Open Collective transaction histories and GitHub commit histories. Our results show that OSS funding should be understood not only by the amount received, but also by how funds are allocated and who receives them. In particular, development-related spending was a major use of funds and was often directed toward existing contributors. We also found that projects allocating funds to development tended to show more favorable commit activity than those that did not. These findings suggest that the sustainability implications of OSS funding depend on spending practices, not merely on the presence of donations.

Acknowledgements This work was supported by JSPS KAKENHI Grant Number JP24H00692, JP25K03102, JP26K02889, and JP26H02500.

References

[1] A. Alami, R. Pardo, and J. Linåker, "Free open source communities sustainability: Does it make a difference in software quality?," Empirical Software Engineering, 2024.

[2] A. Mockus, R.T. Fielding, and J.D. Herbsleb, "Two case studies of open source software development: Apache and mozilla," ACM Transactions on Software Engineering and Methodology, vol.11, no.3, 2002.

[3] O. Nourry, M. Kondo, S. Saito, Y. Iimura, N. Ubayashi, and Y. Kamei, "Myth: The loss of core developers is a critical issue for oss communities," arXiv preprint arXiv:2412.00313, 2024.

[4] J. Zhou, S. Wang, Y. Kamei, A.E. Hassan, and N. Ubayashi, "Studying donations and their expenses in open source projects: a case study of github projects collecting donations through open collectives," Empirical Software Engineering, vol.27, no.1, p.24, 2022.

[5] J. Walden, "The impact of a major security event on an open source project: The case of openssl," 2020 IEEE/ACM 17th International Conference on Mining Software Repositories (MSR), pp.409–419, 2020.

[6] P. Meakpaiboonwattana, W. Tarntong, T. Mekratanavorakul, C. Ragkhitwetsagul, P. Sangaroonsilp, R.G. Kula, M. Choetkiertikul, K. Matsumoto, and T. Sunetnanta, "Social media reactions to open source promotions: Ai-powered github projects on hacker news," 2025 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp.344–355, 2025.

[7] J. Cohen, "A coefficient of agreement for nominal scales," Educational and Psychological Measurement, vol.20, 1960.