Comparing Developer Attraction and Turnover in OSS vs. OSS4SG Projects: An MSR Study

Abstract—**Abstract - August 29, 2024:**

Open source for social good (OSS4SG) projects are projects whose overarching goal focuses on solving a societal issue and driving positive change. Despite the societal importance of such projects and their alignment with the open-source ideology at its very core, there is a clear lack of understanding of the unique characteristics and operational dynamics of open source for social good projects. In this work, we investigate the community dynamics and project characteristics of OSS4SG projects compared to OSS projects. To this end, we leveraged the GitHub API and GraphQL to mine 290 projects (199 OSS4SG, 91 OSS) and answered our main research questions:

- ***RQ1: To what extent do OSS4SG and general OSS projects differ in terms of project characteristics and community dynamics?***
- ***RQ2: How stable are OSS4SG compared to OSS in terms of newcomers and leavers?***
- ***RQ3:* To what extent do turnover and retention of contributors change over time for OSS vs OSS4SG?**

Namely, we studied and compared the rates at which contributors joined and left OSS and OSS4SG projects at different stages of life (early, mid, late). We classified projects as attractive (high join rate and low leave rate), unattractive (low join rate and high leave rate), stable (low join rate and low leave rate), and unstable (high join rate and high leave rate). Our results show that the majority of OSS4SG projects (60%) are stable, while the majority of OSS projects (74%) are unstable. Our results also show that OSS4SG retains more of their contributors and has less turnover. Looking at the join and leave rates for projects across different life stages, we found that the difference is significant when all stages of life and considered together . However, while the difference is significant for the overall lifespans, the difference is far less significant in the late stage of the project's lifespan.

Index Terms—**Open Source, OSS, OSS4SG, companies in open source, motivations, diversity**

I. Introduction

Open Source Software (OSS) has revolutionized the way software can be developed and maintained. They encourage collaboration and foster a community around a project by allowing anyone to contribute, which can also accelerate innovation within a project. OSS project can be a wide range of sizes, from massive global projects to personal pet projects./ They can also be for a range of causes and objectives. This paper is focusing on a smaller, lesser known subset of OSS called OSS4SG OSS4SG stands for Open Source Software for Social Good, and is used to describe any open source project that seeks to benefit society through social causes. in this paper, we want to compare the behaviour, particularity related to contributors, of OSS4SG projects with regular OSS projects.

As open source development continues to grow, understanding the factors that influence project sustainability and contributor retention has become essential. Contributor turnover is a common challenge in OSS, as high turnover rates can disrupt project progress and strain existing contributors. OSS4SG projects, which have emphasis on a social cause, may have different dynamics and patterns for contributor engagement and retention than general OSS projects. By examining these dynamics and the differences between them, we can identify strategies that make OSS4SG communities more resilient or better at retaining contributors, potentially uncovering lessons that could benefit the broader OSS community.

examining OSS4SG alongside OSS and comparing them allows us to better understand how values and objectives of projects impact community dynamics. For example, if OSS4SG projects exhibit higher retention due to their positive social causes and impact, general OSS projects could try to learn strategies from this to improve contributor retention.

This study, not only seeks to fill a gap in OSS and OSS4SG literature, but will also provide actionable insights for OSS project maintainers.

Key Terms SDG: The UN's 17 Social Development Goals OSS: Open Source Software projects that are not in the set of OSS4SG OSS4SG: Open Source Software For Social Good projects are ones that align with one or more of the UN's 17 United Nations Social Development Goals (UN SDGs)

- OSS4SG - Tunover - Join - leave - Attraction - Sticky magnetic

OSS join/ leave rate OSS is challenged in terms of attracting and retaining contributors

OSS4SG (2 paper)

II. Related Work

OSS has become very popular and with its rise of popularity, there has been significant research on turnover, community dynamics, and who contributes to them. However, OSS4SG is much newer and smaller, and has far fewer research papers investigating it and comparing OSS4SG to OSS. This is a gap that we seek to fill with this paper.

This paper introduced the notation of OSS4SG and also compares the contributors of OSS4SG projects and OSS projects. However, this paper focuses on contributor motivation and how they select projects, by interviewing contributors and collecting data. Our paper will build on this by comparing statistical contributor behaviour such as their join and leave rates for OSS4SG and OSS.

This paper investigates turnover in open source projects by examining join and leave rates of core contributors. It gives us a great way to compare turnover in open source projects of different sizes. However since this paper only compares OSS projects and doesn't delve into OSS4SG, our paper can build on their research by using a similar method for comparing OSS4SG to OSS.

III. Research Method

The goal of our study is to explore the differences between OSS4SG (Open Source Software for Social Good) projects and general OSS (Open Source Software) projects. Specifically, we have identified the following research questions following a general to specific approach:

- **RQ1:** To what extent do OSS4SG and general OSS projects differ in terms of project characteristics and community dynamics? **This research question analyses OSS4SG and OSS projects to provide a general comparison of the characteristics of their GitHub repositories. ** **By investigating key metrics and such as project size, commit frequency, issues, and pull requests we can better understand the structural and community differences between OSS4SG and OSS**
- **RQ2:** How stable are OSS4SG compared to OSS projects in terms of the influx of newcomers and the departure of contributors? **This research question performs a deeper analysis of the difference in contributor behaviour, particularly newcomers and departures, between OSS and OSS4SG projects. ** **By investigating the patterns of how often contributors join and leave these projects we can better understand the stability of these projects and their communities over time**
- **RQ3:** How do contributors' attraction and turnover evolve over-time in OSS4SG projects compared to general OSS projects?
  This question expands on RQ2 by further analyzing the contributor behaviour over different stages of a projects life to compare OSS4SG and OSS at different stages. By investigating the evolution of contributor attraction and turnover throughout a project life cycle we can better understand how contributor turnover changes as these projects mature.

A. Dataset and selected projects

We used an existent dataset from Huang et al. [1] The dataset includes a list of 641 OSS projects and 437 OSS4SG projects. To exclude toys / personal projects and ensure that the projects are still active, we followed the approach of Pantiuchina et al. [2]. We included projects that (1) had at least 10 contributors and more than one year of history (2) had at least 500 commits and 50 closed Pull Requests, and (3) were modified at least once in the last year. A Python script was created to automate the project selection process. This inclusion/ exclusion criteria resulted in a total of 289 projects (198 OSS and 91 OSS4SG) that have sufficient commit history and lifespan for our analysis.

TABLE I
The number of projects removed by each filtering criteria

Metric	Overall	OSS4SG	OSS
Number of projects before filtering	**1039**	**422**	**617**
Removed due to low number of contributors	517	122	395
Removed due to low number of commits	131	37	94
Removed due to low number of closed PRs	21	9	12
Removed due to inactivity	81	56	25
Removed due to short lifespan	0	0	0
Number of projects after filtering	**289**	**198**	**91**

TABLE II
List of all info scraped from each Github project

Category	Metrics Scraped
Project Description	- Languages
	- Start Date
	- Last Contribution Date
	- Lifespan
Popularity	- Number of Stars
	- Number of Subscribers
	- Number of Forks
Community	- Number of Contributors
	- Number of Authenticated Contributors
	- Number of Unauthenticated Contributors
	- Number of One-Time Contributors(OTCs)
	- Number of Authenticated OTCs
	- Number of Core Contributors
Contributor Details	- List of Contributors
	- List of Authenticated Contributors
	- List of Core Contributors
	- Number of Commits
Contributions	- Number of Open Issues
	- Number of Closed Issues
	- Number of Open Pull Requests
	- Number of Merged Pull Requests

We used the GitHub API to collect project and community characteristics for each repository up to date (see Table below for the list of characteristics).

An authenticated user is a user making contributions to a repository using their GitHub account. We can make requests for information on authenticated users to get a list of their contributions to the repository which we need. On the other hand, unauthenticated users can still make contributions but there is no account that we can make requests from and therefore they cannot be used for the study. An additional issue for projects with over 500 contributors is that GitHub hides the accounts for all contributors after the 500 with the most contributions, having them appear as unauthenticated regardless of if they or not.Therefore the maximum number of contributors that we can analyze per project is 500.

In order to get all of this information, we wrote python code that makes requests for each repository using the GitHub API. Unfortunately, For each repository, we cannot get all of the desired information with only one request. For each repository we needed to make requests for the actual repository, the contributors, the languages, the closed pull requests, the open pull requests, the merged pull requests, the closed issues, and the commits, for a total of 8 requests per repository.

For each of these repositories, we also collected the following data for each of their authenticated contributors: number of contributions, first contribution date, last contribution date, number of commits in each of the three stages, number of issues in each stage, and number of pull requests in each stage.

To get this contributor information, we needed to use GraphQL since Github API limits does not allow us to get a user's entire commit history. In order to get the information we need using queries, we first had to query for the user ID since you can't just use their username. Then we made three more queries for the commits, issues, an pull requests for a total of 4 queries per contributor.

This led to very long running times for scraping and since we were making so many requests using our token, we would run into rate limit issues. To solve this we would periodically check our current remaining number of requests and rotate to a new token if we were getting close to 0.

The projects in our dataset have varying lifespans, making direct comparisons challenging. As projects evolve, the dynamics of contributor retention and turnover often shift. For instance, projects might experience higher join rates in the early stages, followed by a decline as they mature. To address these differences and capture these life cycle patterns, we divided the projects into distinct life stages based on their maturity level. This approach allows us to analyze and compare projects at similar points in their development. To do this, we used quartile-based binning where the thresholds for dividing the groups are determined using the quartiles of the project's lifespan distribution. The first group includes projects younger than 7 years (below the first quartile). The second group consists of projects aged between 7 years (first quartile) and 9 years (second quartile). The third group covers projects between 9 years (second quartile) and 10 years (third quartile). Finally, the fourth group comprises projects older than 11 years (above the third quartile). This approach allows for a clear distinction between different stages of project maturity, making comparisons more meaningful.

The repositories in the first age group with less than 7 years of life (i.e. the shortest lifespan) were discarded since they did not have any complete stages of life to analyze, making them difficult to compare with other projects.

The second group had a complete early stage to analyze, the third group had both the early and middle stages, and the fourth group had all 3 stages. Since the stages are the same, the early stage from a project from the second group could be compared to the early stage from a fourth group project.

For example, a project with a lifespan of 10 years would fall into group 3, and have an early and middle stage to analyze but would not have a complete late stage

Tables below show the number of projects in each stage and group:

Info was scraped about each contributor in each repository, including first contribution, last contribution, number of commits in each stage, number of issues in each stage, and number of pull requests made in each stage.

TABLE III
DESCRIPTION OF THE DIFFERENT GROUPS OF PROJECTS ANALYZED

Group	Lifespan (years)	Total	OSS4SG	OSS
First Group	0 – 7.421	72	54	18
Second Group	7.421 - 9.115	72	54	18
Third Group	9.115 - 10.923	72	52	20
Fourth Group	>10.923	74	39	35

TABLE IV
DESCRIPTION OF ALL 3 STAGES OF PROJECT LIFE

Stage	Early Stage	Mid Stage	Late Stage
Time period	0 - 7.421	7.421 - 9.115	9.115 - 10.923
Duration	7.421 years	1.694 years	1.808 years
Number of Windows	29	6	7
Total with stage	218	146	74
OSS4SG with stage	145	91	39
OSS with stage	73	55	35

B. Data analysis

To investigate how stable OSS4SG and OSS projects are, we focus on join and leave rates of contributors disaggregated by life stage...

We defined a user's join date as their first contribution, but we can only consider their last contribution to be their leave date if the last contribution date was over 5 months ago to account for users who are still contributors and haven't actually left. (CITE HERE)

We divided each stage of like into 3-month windows for the analysis. Since the stages of life had different lengths, they each had a different number of windows.

Dividing number of joins and number of leaves in a window by the number of total contributors in that window gets us join and leave rates which can be compared to see turnover and retention differences between projects. This method is based on previous work([3].

The join and leave rates for each project is calculated for each window and displayed in a line graph. insert graph here? Then all join and leave rates were averaged out for each window to create only one line for each OSS4SG and OSS to see the difference in contributor turnover.

For each project, the average join rate and leave rate of all windows is calculated. Then this is graphed as a scatterplot, where each point is a project with the X axis showing the leave rate of the project and the Z axis showing the join rate. Using the median join and leave rates for all projects to create lines through the plot, the graph is split into 4 quadrants.

Similar to previous work([3] we define the following quadrants:

- **Attractive (High join rate, low leave rate):** Projects which tend to grow their team. They keep their existing contributors while also attracting new ones. These projects are shown in the upper left.
- **Unattractive (Low join rate, high leave rate):** Projects that struggle with contributor retention and attraction. These projects are shown in the bottom right quadrant.

- **Stable (Low join rate, low leave rate):** Projects that have a steady team with minimal change in contributor numbers. These projects appear in the bottom left quadrant.
- **Unstable (High join rate, high leave rate):** Projects with high turnover, consistently bringing in new contributors but also losing many. These projects are shown in the top right quadrant.

Additionally, some of these categories can be combined to show which projects are "sticky" and "magnetic."

- **Magnetic (High join rate):** Projects that attract a high number of new contributors (combines unstable + attractive).
- **Sticky (Low leave rate):** Projects that retain most contributors, maintaining stability in their teams (combines stable + attractive).

These show the two ways a projects can stay alive, either by retaining its contributors (Sticky) or by attracting new contributors (magnetic)

C. Statistical significance tests

We ran significant tests on the graphed data to see if the difference OSS4SG and OSS join and leave rates were significant. The average join and leave rates per window was tested using the Wilcoxin signed-rank test since the each data point in the list of join rates per window for the OSS projects is paired with the data point for the same window on the OSS4SG list. However, to test the join and leave rates per project (the data shown in the scatterplots), we used the Mann-Whitney U test (also known as the Wilcoxin rank-sum test) since the data were not paired. For each data point in one list, there did not exist and equivalent data point for that project in the other list.

OLD

IV. Findings

RQ1:

RQ2: The first thing we looked at was the scatterplot comparing the average join and leave rate for each project. Each point is a project, with its colour showing whether its an OSS or an OSS4SG and its location showing its join and leave rate.

The grid is split into the four quadrants, as discussed in the methodology section.

TABLE V
THE PERCENT OF OSS AND OSS4SG PROJECTS IN EACH OF THE FOUR QUADRANTS OF THE SCATTERPLOT

Category	OSS4SG	OSS
Attractive	3.4%	1.4%
Unstable	33.8%	74.0%
Stable	60.0%	21.9%
Unattractive	2.8%	2.7%
Sticky	63.4%	23.3%
Magnetic	37.2%	75.4%

A very small percent of both OSS4SG and OSS projects are in the dark green quadrant, in which the projects successfully attract new contributors as well as retain their existing contributors. There are also very few projects in the dark red quadrant, in which projects did not attract new contributors and were losing the contributors they did have.

60% of the OSS4SG projects are in the light green (stable) quadrant, in which projects do not attract new contributors but can keep their current contributors. This is much higher than the 21.9% of OSS projects that are in this quadrant.

74% of the OSS project are in the light red section, in which project attract new contributors but can't retain the contributors they do have. This is much higher than the 33.8% of OSS4SG projects in this quadrant.

The OSS projects are much more unstable and magnetic, while the OSS4SG are more stable and sticky, meaning the OSS4SG retain more of their contributors instead of attracting new ones, while the OSS projects lose more contributors and stay alive by attracting new contributors.

Test Description	Statistic	P-value	Effect Size
Average Join Rates	2714.5	6.03e-09*	-0.3948*
Early Average Join Rates	2570	7.92e-10*	-0.4172*
Mid Average Join Rates	1805.5	0.0063*	-0.2266
Late Average Join Rates	522.5	0.1199	-0.1819

Test Description	Statistic	P-value	Effect Size
Average Leave Rates	2589.5	1.05e-09*	-0.4142*
Early Average Leave Rates	2490.5	2.48e-10*	-0.4296*
Mid Average Leave Rates	1954.0	0.0337*	-0.1763
Late Average Leave Rates	517.5	0.1075	-0.1883

RQ3:

To further investigate this difference in stability between OSS and OSS4SG projects, we then analyzed their join and leave rates over time.

By looking at how the join and leave rates change over time for OSS, and OSS4SG projects, we can see if this difference is for the entire life, if the trend changes over time.

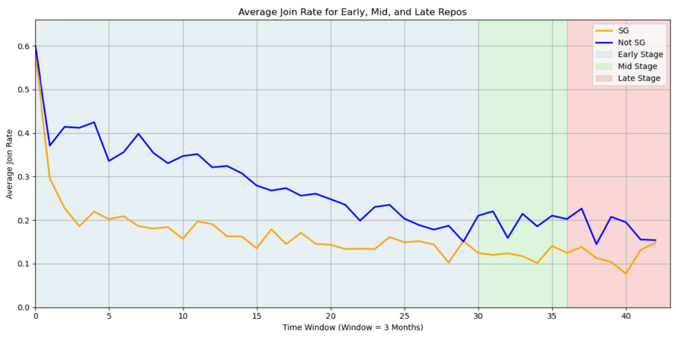

Fig. 1. Line graph of average join rate for all projects over time, using windows of 3 months each

Throughout the early stage, the OSS projects have significantly higher join rates than the OSS4SG projects. This difference is statistically significant when tested with a Wilcoxin test. This agrees with the results of the scatterplot in RQ2. However, in the mid and late stage, this difference becomes much smaller, as OSS projects' join rates decrease to match OSS4SG. This shows that, after OSS projects have been alive for a long time, their contributor join patterns become more similar to OSS4SG.

Test Description	Statistic	P-value	Effect Size
Join Rates	7.0	2.16e-12*	-0.8586**
Early Join Rates	1.0	3.73e-09*	-0.8693**
Mid Join Rates	0.0	0.03125*	-0.8987**
Late Join Rates	0.0	0.015625*	-0.8944**

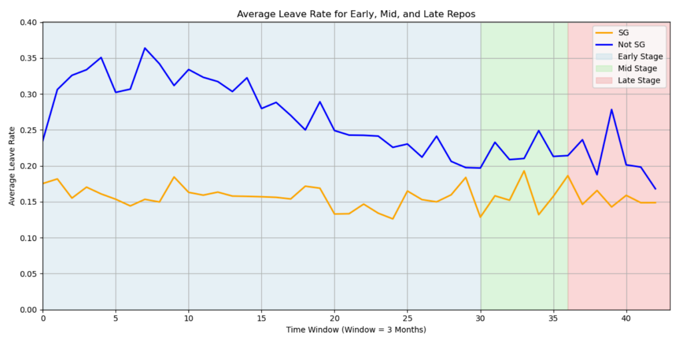

Fig. 2. Line graph of average leave rate for all projects over time, using windows of 3 months each

The leave rates show the same trend: The OSS projects have a much higher leave rate than the OSS4SGs. particularly in the early stage. This also tested to be significant. However, this difference is yet again much smaller in the mid and late stage of the projects' lives.

Test Description	Statistic	P-value	Effect Size
Leave Rates	3.0	5.68e-13*	-0.8656**
Early Leave Rates	0.0	1.86e-09*	-0.8731**
Mid Leave Rates	0.0	0.03125*	-0.8987**
Late Leave Rates	0.0	0.015625*	-0.8944**

Using both of these graphs, it can be seen that OSS projects have higher turnovers(more joins and leaves) in the early stage but their turnover decreases as they get to the late stage where their join and leave rates become more similar to that of the OSS4SGs.

This can be further confirmed by separating the scatter plots from RQ2 into the three stages of life to comparing average join rate and average leave rate per project across different stages.

V. Threats to validity

VI. Discussion

VII. Conclusion

In conclusion, this study highlights key differences in community dynamics and contributor retention between OSS4SG and general OSS projects. Our analysis reveals that OSS4SG projects tend to have higher stability and lower turnover rates than general OSS projects, likely due to the social causes of OSS4SG fostering a more loyal base of contributors. By examining join and leave rates across various stages of project life, we have observed that OSS projects rely more heavily on attracting new contributors , while OSS4SG projects are more effective at retaining their contributors. We also saw that as general OSS projects get later in their life, their community dynamics start to resemble those of OSS4SG projects, possibly due to these projects cultivating loyal contributors, and no longer rely and new contributors. These finding suggest that OSS can become more stable and reduce turnover by learning from OSS4SG.

References

[1] Y. Huang, D. Ford, and T. Zimmermann, "Leaving my fingerprints: Motivations and challenges of contributing to oss for social good," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 2021, pp. 1020–1032.

[2] J. Pantiuchina, B. Lin, F. Zampetti, M. Di Penta, M. Lanza, and G. Bavota, "Why do developers reject refactorings in open-source projects?" *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 31, no. 2, pp. 1–23, 2021.

[3] F. Ferreira, L. L. Silva, and M. T. Valente, "Turnover in open-source projects: The case of core developers," in *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*, 2020, pp. 447–456.