An empirical study of the first contributions of developers to open source projects on GitHub

Vikram N. Subramanian
SWAG Lab
University of Waterloo
vnsubram@edu.uwaterloo.ca

ABSTRACT

The popularity of Open Source Software (OSS) is at an all-time high and for it to remain so it is vital for new developers to continually join and contribute to the OSS community. In this paper, to better understand the first time contributor, we study the characteristics of the first pull request (PR) made to an OSS project by developers. We mine GitHub for the first OSS PR of 3501 developers to study certain characteristics of PRs like language and size. We find that over 1/3rd of the PRs were in Java while C++ was very unpopular. A large fraction of PRs didn't even involve writing code, and were a mixture of trivial and non-trivial changes.

1 INTRODUCTION

Open Source Software (OSS) has always been a vital part of software development - as per the yearly report published by GitHub [4], over 3.6 million repositories depend on only the top 50 OSS projects. These projects couldn't exist without continued contributions from those in the OSS community and therefore it is vital to ensure that new developers are regularly joining these ranks.

While there has been a lot of focus on how to attract and keep OSS developers such as [10][5][11][9][7], there has been little focus on first time OSS contributions on GitHub. Therefore, we try to understand the characteristics of the first contributions made by OSS contributors. Our motivation is to help first timers understand what sort of task they can take up and moderators of OSS projects to understand which tasks they must encourage beginners to take up. We define an open source 'contribution' as a pull request (PR) that has been successfully merged to the parent of a publicly available repository.

Gousios [8] curated GitHub in a dataset called GHTorrent for others to use, but it can't be used to select particular commits/users based on special criteria (such as a user's first OSS PR). Therefore, we develop our own approach based on the GitHub REST API [1] and use that to mine for the data we are interested in. Collecting and analyzing the first contributions of OSS developers is the main contribution of our research.

2 DATA COLLECTION

GitHub follows a 'fork and pull' system where users have to create their own copy of a project (fork a project), make changes, commit them and create a 'PR' to the parent of the forked repository requesting to merge the commits. The moderators of the project and the general public can then review the changes made, request modifications and then the moderators can finally decide to merge the changes or not. Assuming such a frame of reference we follow the below steps to collect the data:

1. We use the REST API[1] to obtain a list of 1000 repositories from The Apache Software Foundation's GitHub page[3] (limited by the API- there are around 1900 repositories in total). We use Apache projects as we wish to study contributors who have contributed to at least one well established project.

2. We collect the usernames of the top 1000 most recent contributors to the above 1000 repositories. We get 15,535 unique users.

3. For every user, we collect a list of all their forked repositories.

4. We then check if the user has made any commits to these repositories. If the user has not, then this repository is dropped.

5. We then check the parent of these repositories and see if commits by the user exist. If it does, we get the chronologically first commit. This commit is possibly the users first open-source contribution on GitHub.

6. Once we collect commits from all the repositories as described in step 5, we sort the commits chronologically and pick the first commit and check to make sure that a PR is associated with this commit. This will be that users first OSS contribution on GitHub. At the end of this step, we get the first PR of 3501 users.

3 RESULTS

3.1 What languages are used in a users first OSS PR?

Motivation: Understanding which languages are popular for first time contributions helps beginners understand which languages they should learn, and educators understand which languages they should teach. It also helps OSS moderators understand which language reaches the most number of first timers.

Approach: The GitHub API lists the files changed by a PR. We used Regex to get only the file extensions. Then we work out the language/file type from the extension. We also account for languages like C++ which can have different file extensions for different files.

Results: Table 1 lists the top 15 languages/file types used. Unsurprisingly, Java is by far the most popular language at 34.67%. This observation aligns with the fact that Java was the most popular language throughout the 2010s [6]. One unexpected observation is that C++ is extremely unpopular and was only the 15th most popular language for first time contributions while it was the 4th

ICSE '20 Companion, May 23–29, 2020, Seoul, Republic of Korea
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7122-3/20/05…$15.00
https://doi.org/10.1145/3377812.3382165

	Language/File Type	No. of files	percentage
1	Java	6242	34.67%
2	Python	1069	5.93%
3	Picture file (PNG)	1067	5.92%
4	Markdown file (MD)	824	4.57%
5	JavaScript	767	4.26%
6	XML	742	4.12%
7	Go	641	3.56%
8	Shell	398	2.21%
9	Scala	385	2.13%
10	JSON	315	1.75%
11	HTML	255	1.41%
12	C	238	1.32%
13	Ruby	194	1.07%
14	Text File	178	0.98%
15	C++	159	0.83%

Table 1: File type used in contributed files

most popular 'main' language(the language with the most bytes of code in that repository) for the OSS repositories we studied.

It is interesting to note that the third most popular file type were image files (PNG). Analyzing commits with PNG files indicates that PNG files are primarily used as a part of documentation, for UIs or in projects where a database of pictures was required. Almost all commits with PNG files had multiple PNG files resulting in the high percentage of PNG file contributions. GitHub uses its own markdown language [2] for documentation (.md files) and it is clearly more popular than any other way of documenting code. Text files come at a distant second with only 0.98% of changes being made in them while 4.57% were made in MD files.

Takeaway: Moderators of OSS projects could use our results to prioritize tasks based on the language of the file that needs to be modified (ideally the most popular current language which is Python and JavaScript) for first time contributors to take on. Developers new to OSS could look into contributing non-code contributions.

3.2 What is the size of contribution in a users first OSS PR?

Motivation: Understanding how many lines of code and files were changed per contributions helps roughly gauge the difficulty of tasks first timers are taking up. OSS moderators can also understand which tasks to allocate for first timers.

Approach: The GitHub API returns the nature of change and number of lines changed for each file in a PR. We got 18,009 files from 3501 PRs. We process that data for our analysis. Non-textual contributions such as adding PNG files are classified as 0 line changes.

Results: Fig 1B describes the number of lines changed. 31.5% of the changes were 1-5 lines and 18.8% of the changes were 6-15 lines with some extreme outliers. The number of files changed per PR follows a distribution similar to the number of lines changed per PR as shown by Fig 1A. Most changes are single file changes with a few extreme outliers that pushes the average to 5.1 changed files. File changes are classified as modified, added, renamed or removed by GitHub. 54.77% of all files changed were modifications, 35.41% were added, 5.85% were renamed and 3.94% were removed.

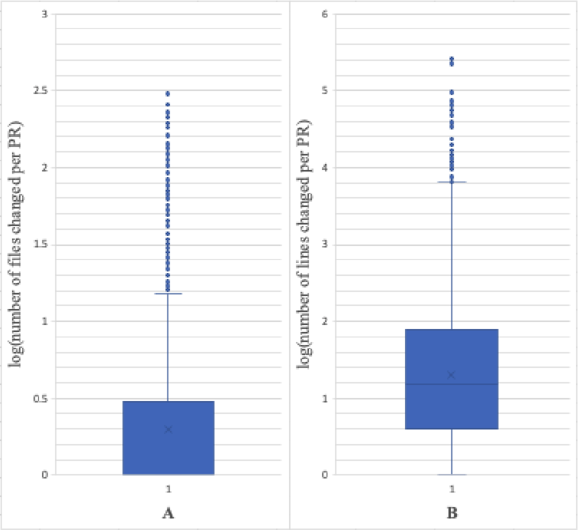

Figure 1: A- Box plot of the log of number of files changed per PR. B- Box plot of the log of number of lines changed per PR.

It is interesting to note that contributions with multiple file changes are usually when new files are created to initialize a new feature/aspect of that project. This accounts for a large percentage of the 35.41% of files that were 'added'.

Takeaway: First time contributors should not be discouraged to take up big tasks as the entire 4th quartile of PRs studied were multifile and change over 75 lines. However, with a median change of 15 lines and over half the contributions being only modifications, it is clear that most first timers take up small to medium sized changes such as bug fixes and documentation edits.

4 LIMITATIONS AND THREATS

Some limitations of the method used to mine data are-

1.The GitHub API limits the number of data units returned per request to 1000. This means that if any of the queries made above have more than 1000 data units, then that user will have to be dropped. Here we define a data unit as the smallest unit of data the API returns for a request. The limit problem exists even in popular Github datasets like GHTorrent too.

2.The user has complete control over which repositories they want to list under their 'owned' repositories section and therefore if the user has decided to remove their first contribution from their page, then this script will not pick up that contribution. However, we know of no reason why a developer would do so.

3.OSS projects have existed long before GitHub- our approach picks up only a user's first GitHub OSS contribution.

5 FUTURE WORK

We are currently working on a qualitative and more in-depth study of our data- we are classifying commits into types (such as bug fix, new feature, documentation etc.) and studying the nature of contributions made. We will also be studying the common characteristics of repositories that first time OSS contributions are made to. The scripts used to mine the data and the data itself is available at https://bit.ly/2Zzxk5i

REFERENCES

[1] [n.d.]. GitHub API v3. https://developer.github.com/v3/

[2] [n.d.]. Mastering Markdown. https://guides.github.com/features/mastering-markdown/

[3] 2019. The Apache Software Foundation. https://github.com/apache

[4] 2019. The State of the Octoverse. https://octoverse.github.com/

[5] Sogol Balali, Igor Steinmacher, Umayal Annamalai, Anita Sarma, and Marco Aurélio Gerosa. 2018. Newcomers' Barriers. . . Is That All? An Analysis of Mentors' and Newcomers' Barriers in OSS Projects. *Computer Supported Cooperative Work (CSCW)* 27 (2018), 679–714.

[6] S. Cass. 2014. The top 10 programming languages spectrum's 2014 ranking [dataflow]. *IEEE Spectrum* 51, 7 (July 2014), 68–68. https://doi.org/10.1109/MSPEC.2014.6840816

[7] Felipe Fronchetti, Igor Scaliante Wiese, Gustavo Pinto, and Igor Steinmacher. 2019. What Attracts Newcomers to Onboard on OSS Projects? TL;DR: Popularity. In *OSS*.

[8] Georgios Gousios. 2013. The GHTorent Dataset and Tool Suite *(MSR '13)*. IEEE Press, 233–236.

[9] Igor Steinmacher, Tayana Uchoa Conte, Christoph Treude, and Marco Aurélio Gerosa. 2016. Overcoming Open Source Project Entry Barriers with a Portal for Newcomers *(ICSE '16)*. Association for Computing Machinery, New York, NY, USA, 273–284. https://doi.org/10.1145/2884781.2884806

[10] I. Steinmacher, C. Treude, and M. A. Gerosa. 2019. Let Me In: Guidelines for the Successful Onboarding of Newcomers to Open Source Projects. *IEEE Software* 36, 4 (July 2019), 41–49. https://doi.org/10.1109/MS.2018.110162131

[11] Igor Steinmacher, Igor Scaliante Wiese, Tayana Conte, Marco Aurélio Gerosa, and David Redmiles. 2014. The Hard Life of Open Source Software Project Newcomers *(CHASE 2014)*. Association for Computing Machinery, New York, NY, USA, 72–78. https://doi.org/10.1145/2593702.2593704