4-29-2022

Alternative Participation: An exploratory study on participation in open source software beyond code contributions

Benjamin Dow
bmd8811@rit.edu

RIT

Alternative Participation: An exploratory study on participation in open source software beyond code contributions

by

Benjamin Dow

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Software Engineering

Department of Software Engineering
Golisano College of Computing and Information Sciences

Rochester Institute of Technology
Rochester, NY
April 29, 2022

Approval Committee Members

Dr. J Scott Hawker	Director of Graduate Program Committee Member
Samuel Malachowsky	Primary Advisor Committee Member
Dr. Christian Newman	Committee Member

Alternative Participation: An exploratory study on participation in open source software beyond code contributions

Benjamin Dow
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14623
Email: bmd8811@rit.edu

***Abstract*—While it is commonly accepted that well-engineered commercial software projects rely on a variety of activities, this not the case with open source software development. A poorly understood area of open source software is what types of activities are present and being completed day-to-day. Understanding what activities exist would give developers and project leads additional attributes of the software engineering process to modify and improve. Identifying these activities is challenging as they are often abstract in nature and activities may not be formally defined within projects, but may still be executed; for example, the way in which a project accepts feedback may be defined or simply accepted through the issue tracker with no formal declaration. In this paper I investigate alternative participation activities in a variety of open source software projects. I found that a majority of these projects have alternative participation occurring but many struggle to formally define discrete activities or provide calls to action.**

1. Introduction

Open source software is an incredibly important and impactful part of the software engineering industry. As with any well-engineered software product, there are a number of activities that are completed beyond the implementation related steps for a project. Any number of relevant activities will be completed for a software project, from project management and product planning to architecture design and creation of documentation artifacts.

This paper describes a study performed on a set of open source software projects to explore the activities and roles beyond code contributions that are being performed. The motivation for this study comes from a desire give more tools to project administrators and maintainers to further refine the engineering process for open source projects.

In this paper I will first lay out the research objectives as well as the four main research questions guiding this study. Next, I will outline related works and identify the similarities and differences to this study. I will then describe the method with which this study was performed, then the initial results. Following the initial results from this study, I will discuss what trends exist and address each of the research questions with relation to the dataset. Finally, I will list the major limitations of this study and what future work could look like in this area.

2. Research Objectives

The objective of this research is to identify the activities that exist in open source software projects, apart from implementation. The goal is to understand what activities already exist, the amount participation (if data is available), and how projects make these activities known to their communities.

3. Research Questions

To guide the research of this study, I have defined four essential research questions with regard to alternative roles and activities in open source software.

RQ1: Do open source software projects have distinct roles or activities that exist beyond implementation?

This research question has the goal of understanding to what extent alternative roles and activities play in open source software already, and will provide a basis for the questions that follow. Essentially this research question seeks to understand how common non-implementation alternative activities already are in open source software projects.

RQ2: What common participation activities repeat across open source projects?

Understanding the patterns that exist across open source projects is an essential part of this study. Just

like there exists a number of non-implementation roles in commercial software engineering (Project Management, Documentation, Product Management, etc), this question theorises that a number of activities will be reoccurring across open source software projects as well. This question, for the sake of this study, will also include activities that have unique, and specific purposes which warrant further investigation.

RQ3: Do open source software projects formalize alternative participation activities?

In open source software, being that the community is so important to project success, tasks and development activities are often formalized, laid out, and provided as an example for where individuals can start contributing. This third research question seeks to understand if there is a similar formalization of roles and activities for non-implementation tasks, such as the ones that may have been identified in RQ2. An understanding of formalization of non-implementation roles and activities will likely give some insight to open source project priorities and processes as related to the roles and activities potentially found within the project.

RQ4: Do open source projects track contributions other than code and what level of participation do they have?

Open source software, as an artifact of being hosted on development-focused websites (GitHub, Source Forge, etc), almost universally tracks contributions to implementation tasks. This question exists to understand if a similar approach is taken to activities that fall outside the realm of implementation. For the projects that do have this data, this question also seeks to understand the participation level in non-implementation roles and activities.

4. Related Works

Much of the research into roles within open source software has focused on particular developer roles and the evolution, transformation, and effect on sustained participation they have [1], [6], [7]. These studies have examined roles largely related to code contribution, but also at roles for leadership in a project, such as active maintainers. Developers in these roles take on responsibilities beyond just code contribution, such as managing the project as a whole, and helping make bigger picture decisions. This paper will focus on the participation in those activities and roles *beyond* core maintainers, and how those in the community of the project and open source can contribute beyond code.

There have been a number of studies done on the barriers faced by newcomers to open source software, particularly focusing on making their first commit to the project [4]. A number of these studies have looked at social barriers in particular and how to overcome them [3], [4], [5]. These studies give common reasons newcomers are unable, or find it difficult to participate in open source software. This study does not look at alternative participation roles with respect to newcomer barriers, however alternative participation roles may be a solution to overcoming some of these barriers.

5. Method

In this study I perform a qualitative and quantitative analysis on a set of data collected from 46 open source project repositories. A qualitative approach is used to understand the types of participation occurring in open source projects, as well as to understand where that participation is being driven from. A quantitative approach is used to understand the breadth of participation in activities being undertaken, as well as the depth of participation within each activity.

5.1. Selecting Candidate Repositories

The projects and repositories for this study were located using GitHub's API for searching repositories based on certain criteria. As such, all the projects and repositories in this study use GitHub as their primary tool for storing code and other project based artifacts. The following criteria were used to filter and select the repositories for this study.

- More than 45 contributors
- Less than 400 contributors
- A commit or release in the six months before February 3rd 2022
- More than 500 Stars (Favorites/Bookmarks)

Each of these criteria are quantifiable and filterable with the GitHub API, except for the number of contributors on a project. With the GitHub API this data is not accessible by those without "push" permissions on the repository. Each repository was also viewed manually and checked to ensure that it fit other qualitative criteria identifiable by the program used to narrow down repositories. Originally, 50 repositories were randomly selected selected for the study. However, during manual data extraction, four were eliminated. The four that were eliminated fit into categories not suitable for this study, such as example/teaching projects, information repositories, or other non-software projects.

5.2. Data Extraction

Once the 46 repositories for this study were identified, each repository was manually examined with respect to the following questions.

Automatically Collected Data

- Name
- URL
- Repository Creation Date
- Number of Stars
- Resource Path
- Last Release
- Last Update/Commit Push

Project Structure

- Does the project contain a "How to Contribute" section?
- Does the issue tracker have tags for non-implementation tasks?
- Does the project track non-implementation contributions?
- Does the project contain a public project roadmap?

Activity Identification

- What activities exist?

For Each Activity Identified

- Is this an implementation activity?
- What evidence supports the existence of this activity?
- Does this activity have a formal "call to action?"
- Is this activity defined in a "how to contribute" artifact?
- What participation/contribution data exists for this activity?

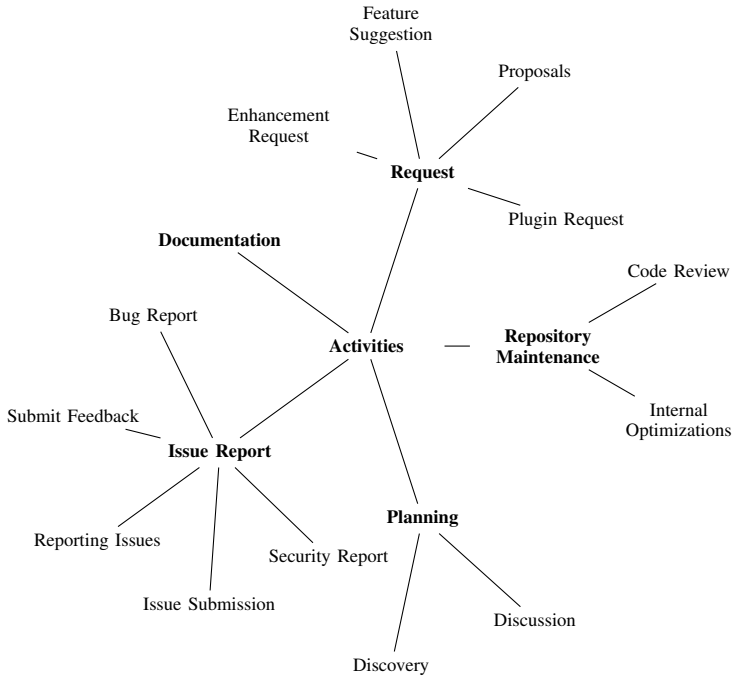
For each repository, the labels used in issue trackers/pull requests were also collected. The intention behind collecting this data is two-fold. The first intention is to find labels, using basic text comparisons, that relate to alternative participation activities and compare the existence of such labels to activities found manually. The second intention is as data for participation; data contained within these labels will give insight into some of the participation in alternative activities.

5.3. Categorical Simplification

After data extraction and creating a list of activities found within the repositories, a natural step was to categorize like activities. With this set of repositories and similar activities, five categories became apparent; these categories and their associated activities are shown in Figure 1.

5.3.1. Issue Report. The common theme of activities sorted into Issue Report are activities that encourage the use of GitHub's issue tracker or other method to submit reports, feedback, or other items that relate directly to the project and items that are currently in scope. Issue Report activities are for project community members to give direct feedback to the current state of the project, and give improvement suggestions that are related to the feature set and current scope.

Figure 1: Activity Categories



5.3.2. Request. Activities sorted into Request have action items relating to aspects in the project that are not currently in scope or implemented. Request activities are ways for individuals to involve themselves in the project through suggesting ways (in their opinion) to improve or continue the project.

5.3.3. Planning. Planning activities are such that individuals in the project community can join the development process of the project without writing code, yet still have some impact on the final architecture/design. Planning activities are the intermediate step between development and the request activities discussed above.

5.3.4. Documentation. Documentation is a very self-explanatory category. Each of the activities sorted in this category directly relate to the writing or improvement of the project documentation.

5.3.5. Repository Maintenance. Activities that fall under the scope of Repository Maintenance deal with the maintenance of or improvement of the repository beyond implementation. These activities may improve code quality, workflow, or processes.

6. Results

The resulting data that was collected was dense and was processed in a variety of ways to give some insight into the patterns and trends that exist. The first step in that endeavor was categorizing the accumulated activities as discussed in Section 5.3. The data was further

TABLE 1: Dataset Statistics

# of Repositories Surveyed	46
# of Repositories with Activities	28
% of Repositories with Activities	60.9%
# of Unique Activities	13
# of Repositories with "How to Contribute" Section	28
% of Repositories with Alternative Activity label	84.7%

TABLE 2: Activity Categories

Category	% of Activities	% of Projects with at least one activity in category
Issue Report	37.8%	36%
Documentation	28.9%	27.3%
Request	22.2%	21.7%
Planning	6.7%	6.5%
Repository Maintenance	4.4%	4.4%

investigated for simple percentages, counts, and other basic characteristics that represent the trends present. The full dataset is available on GitHub [2].

6.1. Dataset Statistics

A set of general, simple statistics were generated for the dataset. To give context to future calculations and statistics, these numbers focus on the research questions outlined in Section 3. Table 1 lists these statistics. For RQ1, the number of repositories surveyed, number of repositories with activities, and percentage of repositories with activities identified are listed. To focus on RQ2 the number of unique activities was calculated. The number of repositories with "How to Contribute Section" focuses on RQ3, and the percent of repositories with alternative activity label gives context to RQ4.

6.2. Activity Categories

Table 2 outlines two simple figures relating to the activities contained within categories. The first is the percentage of total activities identified that were sorted into each category. The second is the number of projects that have at least one activity per category. These two simple characteristics of the activity/category data give some understanding as to the spread and frequency of activities and activity categories within this dataset.

6.3. Calls to Action

Table 3 gives insight to the commonality of projects making "calls to action" for the activities/category. These percentages were calculated by looking at the activities identified and the data collected about their corresponding repository having calls to action or listing the activity in a how to contribute section. The percentages then represent how many activities in each category fit the description of having a call to action or having the activity listed in how to contribute.

TABLE 3: Calls to Action by Activity Category

Category	% with Call to Action	% in "How to Contribute"
Issue Report	70.59%	58.8%
Documentation	46.15%	38.5%
Request	50.0%	40.0%
Repository Maintenance	50.0%	50.0%
Planning	0.0%	0.0%

TABLE 4: Issue Labels by Category

Category	% of Total Projects With Label in Category	% With Participation in Category
Report	93.5%	100%
Documentation	54.4%	100%
Request	91.3%	100%

6.4. Participation

Table 4 focuses on the participation data available and collected via repository labels. Three type of labels were identified to correspond to three categories of activities. The numbers found represent the number of projects that have a label belonging to the category, as well as the number of projects with participation in the category. These two numbers start to give some idea as to what level of participation is occurring in alternative activities.

7. Discussion

RQ1: Do open source software projects have distinct roles or activities that exist beyond implementation?

The data set collected from the repositories surveyed indicates that 60.9% of repositories surveyed had at least one activity, beyond code contributions, having been identified by manual repository review. 84.8% of repositories contained a label in their issue tracker that fell into a category of alternative activities.

RQ2: What common participation activities repeat across open source projects?

There was a very strong pattern of repeating activities throughout the dataset. In particular there were five

categories that each of the activities identified could be classified as; this is shown in Table 2. The most common category for activities to fall under was Issue Report, followed by Documentation and Request, then Planning, and finally Repository Maintenance. This order remains true when looking at the number of repositories that have at least one activity in each category. It makes intuitive sense that the category with the highest number of activities would also correlate with what categories repositories have activities in.

Common Trends. One of the most common activities identified was labeled Issue Submission (Issue Report Category). The most common use case of this activity was for individuals to submit "Issues" to the project. "Issues" could include problems or bugs, feature enhancements (different from feature requests), or any number of other items. The common theme was that these "Issues" generally related to feature that were already implemented or in progress.

Unique Activity: Security Report. The Axios repository had an activity for security reports which has the intentions for individuals to make the maintainers aware of security vulnerabilities within the project. Axios, a project dealing with HTTP requests, has a particular interest in security vulnerabilities. The outline for this activity is defined in a file called SECURITY.md and has instructions for who to email with security vulnerabilities so they don't end up in the general issue tracker (thus revealing the vulnerability to the public) and can be fixed before they are a wider problem. This was the only instance within the dataset of a security related activity.

RQ3: Do open source software projects formalize alternative participation activities?

Table 3 presents data with regard to calls to action by activity category. Unsurprisingly, the rate at which repositories have calls to action for a particular category of activity appears to correlate with the rate at which activities are categorized as such in Table 2. It is also interesting that for categories with the highest rate of calls to action (Issue Report, Documentation and Request), there is a range of 7.7% to 11.8% in difference from the percent of repositories with calls to action. This means that in the categories of the highest call to action rate, a significant portion are not putting these calls to action in a recognized "How to Contribute" document.

RQ4: Do open source projects track contributions other than code and what level of participation do they have?

The data collected with regards to participation has limited, but interesting insights. Simple text analysis was used to identify labels within projects that relate to one of three categories: Reports, Documentation, or Request. These correspond to the categories identified earlier with regards to activities identified. Table 4 highlights a portion of this data. Interestingly, for each category, if a project had labels in a category, there was some level of participation occurring within that category. This does not mean that every label within the category had participation, rather at least one label had some level of participation. This could be indicative of projects that have put in the effort to create labels for various activities or categories do so with the intention to use them, though it does not necessarily mean that the intention was to give place to a specific activity or category of activity. There may also be labels for alternative participation activities that fall out side of the categories defined and the bounds of the text analysis that was performed that are not reflected in this data.

8. Limitations & Future Work

The goal of the study described in this paper was an exploration of alternative participation activities within open source software and a look at a smaller number of repositories with a wider breadth of questions. As such there were a number of areas in this study that have opportunities for further elaboration in focused future work.

8.1. Participation

A large limitation of this study was the amount of data that was collected and analyzed with regards to real world participation in activities. A very limited dataset of labels in repositories and their corresponding use in the repositories issue tracker was collected. Because of this, little understanding to trends can be gained. A possible area for future work to explore would be the participation patterns within repositories containing alternative participation activities.

8.2. Label Text Analysis

Repository Maintenance and Planning were omitted from Table 4 as it was not practical to (manually or through text analysis,) identify label tags that would correspond to the appropriate activity category. Opportunity for future work in this area exists to analyze and more deeply understand the usage of GitHub repository labels with respect to alternative participation categories.

8.3. "How to Contribute" Documents

This study takes a brief look into the prevalence of activity calls to action within repositories and whether these calls to action exist within established and recognized how to contribute documents. Though this study

makes observations about activity calls to actions having a lower rate in appearing in these documents, it does not dive into how effective that may be. Future work in this area could seek to identify what an effective How to Contribute document looks like with regards to alternative participation activities.

9. Conclusion

This study was meant as an exploratory look into activities beyond code contributions within open source software projects hosted on GitHub. The data collected by this study indicates the existence of these activities. The activities identified had a number of attributes in-common and were thus categorized into groups by these attributes and common traits. These categories were named: Issue Report, Documentation, Request, Repository Maintenance, and Planning. The most popular category, Issue Report, contained 37.8% of the activities identified with a total of 17 out of 46 projects having at least one activity in this category. Call to actions for the activity categories correlated with the number of activities within each category. Participation data, while limited, showed a trend towards use of labels if they existed in a repository. This study shows that there are a number of areas to be explored within alternative participation activities that would give insight as to how individuals can interact and participate in open source projects beyond code contributions.

References

[1] Can Cheng, Bing Li, Zeng-Yang Li, Yu-Qi Zhao, and Feng-Ling Liao. Developer role evolution in open source software ecosystem: An explanatory study on gnome. *Journal of Computer Science and Technology*, 32(2):396–414, 2017.

[2] Benjamin Dow. Alternative Participation Roles Data and Analysis, 4 2022. This repository contains all of the data collected and code used to analyze it for this paper github.com/bendow/AlternativeParticipationRolesDataCollection.

[3] Christopher Mendez, Hema Susmita Padala, Zoe Steine-Hanson, Claudia Hildebrand, Amber Horvath, Charles Hill, Logan Simpson, Nupoor Patil, Anita Sarma, and Margaret Burnett. Open source barriers to entry, revisited: A sociotechnical perspective. In *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, pages 1004–1015, 2018.

[4] Igor Steinmacher, Tayana Conte, Marco Aurélio Gerosa, and David Redmiles. Social barriers faced by newcomers placing their first contribution in open source software projects. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work amp; Social Computing*, CSCW '15, page 1379–1392, New York, NY, USA, 2015. Association for Computing Machinery.

[5] Igor Steinmacher, Marco Gerosa, Tayana U. Conte, and David F. Redmiles. Overcoming social barriers when contributing to open source software projects. *Computer Supported Cooperative Work (CSCW)*, 28(1-2):247–290, 2018.

[6] M.R. Martinez Torres, S.L. Toral, M. Perales, and F. Barrero. Analysis of the core team role in open source communities. In *2011 International Conference on Complex, Intelligent, and Software Intensive Systems*, pages 109–114, 2011.

[7] Zhongjie Wang and Dewayne E. Perry. Role distribution and transformation in open source software project teams. In *2015 Asia-Pacific Software Engineering Conference (APSEC)*, pages 119–126, 2015.

Url	Name	# Of Contributors	Resource Path	Stars	Last Push	Latest Release	updatedAt
https://github.com/vantage-sh/ec2instances.info	ec2instances.info	107	/vantage-sh/ec2instances.info	3675	2022-01-18T23:18:18Z		2022-01-30T15:01:07Z
https://github.com/webpack-contrib/webpack-bundle-analyzer	webpack-bundle-analyzer	56	/webpack-contrib/webpack-bundle-analyzer	11566	2022-01-22T12:10:12Z		2022-01-30T18:07:26Z
https://github.com/kentcdodds/advanced-react-hooks	advanced-react-hooks	51	/kentcdodds/advanced-react-hooks	1548	2022-01-04T00:19:33Z		2022-01-30T14:19:29Z
https://github.com/jessepollak/card	card	77	/jessepollak/card	11393	2022-01-25T12:48:08Z	2021-08-14T12:11:32Z	2022-01-27T17:53:12Z
https://github.com/brettwooldridge/HikariCP	HikariCP	115	/brettwooldridge/HikariCP	16299	2022-01-19T21:39:50Z		2022-01-30T15:39:30Z
https://github.com/coreybutler/nvm-windows	nvm-windows	61	/coreybutler/nvm-windows	19314	2022-01-18T15:19:18Z	2021-12-10T19:16:12Z	2022-01-30T21:46:59Z
https://github.com/streamlink/streamlink	streamlink	246	/streamlink/streamlink	7044	2022-01-25T18:35:51Z	2022-01-25T18:42:59Z	2022-01-30T20:58:38Z
https://github.com/wurstmeister/kafka-docker	kafka-docker	70	/wurstmeister/kafka-docker	5956	2021-12-30T16:18:35Z		2022-01-30T08:06:25Z
https://github.com/alibaba/Sentinel	Sentinel	163	/alibaba/Sentinel	18356	2022-01-27T03:23:27Z	2021-12-31T07:11:23Z	2022-01-30T15:24:04Z
https://github.com/nikic/PHP-Parser	PHP-Parser	115	/nikic/PHP-Parser	15183	2022-01-25T23:13:32Z	2021-11-30T19:40:18Z	2022-01-30T20:54:33Z
https://github.com/axios/axios	axios	329	/axios/axios	90798	2022-01-29T14:22:19Z	2022-01-18T07:28:27Z'	2022-01-30T21:47:39Z
https://github.com/MyCATApache/Mycat-Server	Mycat-Server	99	/MyCATApache/Mycat-Server	9217	2022-01-19T13:27:04Z	2020-11-02T09:46:31Z'	2022-01-28T21:54:22Z
https://github.com/ctimmerm/axios-mock-adapter	axios-mock-adapter	64	/ctimmerm/axios-mock-adapter	2874	2022-01-14T06:30:40Z	2021-08-13T06:30:13Z	2022-01-29T11:30:02Z
https://github.com/jmschrei/pomegranate	pomegranate	60	/jmschrei/pomegranate	2820	2022-01-04T19:13:10Z	2016-03-30T20:01:37Z	2022-01-29T16:50:21Z
https://github.com/standard/standard	standard	170	/standard/standard	26582	2022-01-28T12:04:34Z		2022-01-30T18:09:20Z
https://github.com/cookiecutter/cookiecutter	cookiecutter	249	/cookiecutter/cookiecutter	16130	2022-01-27T22:31:00Z	2021-05-14T09:48:19Z	2022-01-30T19:58:14Z
https://github.com/sveltejs/sapper	sapper	122	/sveltejs/sapper	7182	2022-01-22T00:25:35Z		2022-01-30T16:59:37Z
https://github.com/firebase/FirebaseUI-Android	FirebaseUI-Android	96	/firebase/FirebaseUI-Android	4358	2022-01-03T16:06:24Z	2021-07-30T14:10:30Z	2022-01-30T11:37:53Z
https://github.com/arsduo/koala	koala	96	/arsduo/koala	3505	2022-01-25T22:42:10Z		2022-01-24T16:43:14Z
https://github.com/benedmunds/CodeIgniter-Ion-Auth	CodeIgniter-Ion-Auth	206	/benedmunds/CodeIgniter-Ion-Auth	2295	2022-01-09T04:43:03Z		2022-01-30T01:19:08Z
https://github.com/tpope/vim-rails	vim-rails	91	/tpope/vim-rails	3897	2022-01-16T17:14:23Z		2022-01-29T11:11:12Z
https://github.com/rxhanson/Rectangle	Rectangle	86	/rxhanson/Rectangle	15229	2022-01-21T01:05:15Z	2022-01-07T03:25:42Z	2022-01-30T18:28:48Z
https://github.com/reactstrap/reactstrap	reactstrap	249	/reactstrap/reactstrap	10231	2022-01-25T22:50:29Z	2021-11-12T21:53:43Z	2022-01-30T19:36:57Z
https://github.com/symfony/security-guard	security-guard	47	/symfony/security-guard	1313	2022-01-28T11:54:58Z	2022-01-28T11:58:46Z	2022-01-28T16:46:13Z
https://github.com/geohot/tinygrad	tinygrad	58	/geohot/tinygrad	5154	2022-01-17T18:44:41Z		2022-01-29T22:26:35Z
https://github.com/aws/chalice	chalice	157	/aws/chalice	8616	2022-01-30T18:13:32Z		2022-01-30T16:36:24Z
https://github.com/mybatis/mybatis-3	mybatis-3	194	/mybatis/mybatis-3	16740	2022-01-27T18:36:25Z	2021-12-25T19:16:08Z	2022-01-30T15:23:14Z
https://github.com/thephpleague/oauth2-client	oauth2-client	185	/thephpleague/oauth2-client	3197	2022-01-05T17:08:20Z	2021-12-22T16:44:43Z	2022-01-29T17:56:55Z
https://github.com/fent/node-ytdl-core	node-ytdl-core	80	/fent/node-ytdl-core	3065	2022-01-16T13:25:55Z	2022-01-11T13:11:10Z	2022-01-30T17:36:39Z
https://github.com/feross/thanks	thanks	86	/feross/thanks	2760	2022-01-28T05:25:17Z		2022-01-24T04:09:11Z
https://github.com/sharkdp/hyperfine	hyperfine	65	/sharkdp/hyperfine	10170	2022-01-03T21:58:10Z	2021-10-17T17:47:28Z	2022-01-30T20:24:09Z
https://github.com/igorescobar/jQuery-Mask-Plugin	jQuery-Mask-Plugin	46	/igorescobar/jQuery-Mask-Plugin	4554	2022-01-04T05:21:07Z		2022-01-29T11:16:09Z
https://github.com/athityakumar/colorls	colorls	55	/athityakumar/colorls	3939	2022-01-18T23:11:00Z	2021-03-10T20:54:58Z	2022-01-30T06:53:14Z
https://github.com/katspaugh/wavesurfer.js	wavesurfer.js	206	/katspaugh/wavesurfer.js	5981	2022-01-20T12:55:34Z	2021-08-16T13:54:17Z	2022-01-29T06:33:19Z
https://github.com/jacomyal/sigma.js	sigma.js	54	/jacomyal/sigma.js	9621	2022-01-26T17:20:24Z	2022-01-26T17:19:29Z	2022-01-30T19:59:54Z
https://github.com/styled-components/styled-components	styled-components	295	/styled-components/styled-components	35822	2022-01-24T21:29:58Z	2021-10-19T13:37:20Z	2022-01-30T20:49:58Z
https://github.com/apache/echarts	echarts	162	/apache/echarts	49553	2022-01-29T09:05:00Z	2022-01-26T17:57:56Z	2022-01-30T16:09:07Z
https://github.com/capnproto/capnproto-rust	capnproto-rust	59	/capnproto/capnproto-rust	1282	2022-01-08T14:31:15Z		2022-01-30T04:53:10Z
https://github.com/PaddlePaddle/PaddleHub	PaddleHub	55	/PaddlePaddle/PaddleHub	7481	2022-01-21T02:49:57Z	2021-04-16T08:20:11Z	2022-01-30T07:38:15Z
https://github.com/Siccity/xNode	xNode	51	/Siccity/xNode	2145	2021-12-26T20:12:51Z	2020-05-29T20:24:53Z	2022-01-30T18:59:10Z
https://github.com/trailofbits/algo	algo	160	/trailofbits/algo	24514	2022-01-29T22:32:26Z	2019-07-31T15:45:16Z	2022-01-30T21:14:57Z
https://github.com/sass/libsass	libsass	112	/sass/libsass	4263	2021-12-26T07:27:31Z	2021-05-21T00:57:44Z	2022-01-28T09:33:11Z
https://github.com/iawia002/lux	lux	76	/iawia002/lux	16875	2022-01-27T02:24:10Z	2022-01-13T05:09:36Z	2022-01-30T17:44:13Z
https://github.com/go-resty/resty	resty	75	/go-resty/resty	5567	2021-12-21T04:07:49Z	2021-11-04T05:33:30Z	2022-01-28T15:23:16Z
https://github.com/pydata/pandas-datareader	pandas-datareader	75	/pydata/pandas-datareader	2220	2022-01-07T17:48:19Z	2021-07-13T12:27:15Z	2022-01-29T13:53:00Z
https://github.com/php-pm/php-pm	php-pm	70	/php-pm/php-pm	6372	2022-01-09T11:11:11Z	2022-01-09T11:11:11Z	2022-01-30T14:31:27Z

Table 5: Repositories