A Dataset of Enterprise-Driven Open Source Software

Diomidis Spinellis  
Zoe Kotti  
Konstantinos Kravvaritis  
Georgios Theodorou  
Panos Louridas  
{dds,zoekotti,kravvaritis, louridas}@aueb.gr  
Athens University of Economics and Business

ABSTRACT

We present a dataset of open source software developed mainly by enterprises rather than volunteers. This can be used to address known generalizability concerns, and, also, to perform research on open source business software development. Based on the premise that an enterprise’s employees are likely to contribute to a project developed by their organization using the email account provided by it, we mine domain names associated with enterprises from open data sources as well as through white- and blacklisting, and use them through three heuristics to identify 17,264 enterprise GitHub projects. We provide these as a dataset detailing their provenance and properties. A manual evaluation of a dataset sample shows an identification accuracy of 89%. Through an exploratory data analysis, we found that projects are staffed by a plurality of enterprise insiders, who appear to be pulling more than their weight, and that in a small percentage of relatively large projects development happens exclusively through enterprise insiders.

CCS CONCEPTS

• Software and its engineering → Open source model; • Social and professional topics → Computing and business; • General and reference → Empirical studies.

KEYWORDS

Software engineering economics, software ecosystems, open source software in business, Fortune Global 500, SEC 10-K, SEC 20-F, EDGAR, dataset

ACM Reference Format:

Diomidis Spinellis, Zoe Kotti, Konstantinos Kravvaritis, Georgios Theodorou, and Panos Louridas. 2020. A Dataset of Enterprise-Driven Open Source Software. In 17th International Conference on Mining Software Repositories (MSR '20), October 5–6, 2020, Seoul, Republic of Korea. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3379597.3387495

1 INTRODUCTION

Despite the size and wealth of software product and process data available on GitHub, their use in software engineering research can be problematic [9, 28], raising issues regarding the generalizability of the corresponding findings [47]. In particular, the open source nature of accessible GitHub repositories means that projects developed by volunteers through open source software development processes [12, 42] are overrepresented, biasing results, especially those related to software architecture or communication and organization structures, through the application of Conway’s Law [8, 24]. In addition, many researchers are investigating differences between open source and proprietary software products and processes [3, 33, 38, 43].

Here we present a dataset of open source software developed mainly by enterprises rather than volunteers. This can be used to address the identified generalizability concerns and, also, to perform research on the differences between volunteer and business software development. One might think that open source software development by enterprises is a niche phenomenon. As others have identified [40] and also as is evident from our dataset, this is far from true. A series of queries on GitHub PushEvents published during 2017 found that companies such as Microsoft and Google had hundreds of employees contributing to open source projects [26].

The goal of the dataset’s construction is to create a set of GitHub projects that are most probably developed by an enterprise. For the purposes of this work, we define as an enterprise project, one that is likely to be mainly developed by financially compensated employees, working full time under an organization’s management. This definition excludes volunteer efforts such as Linux, KDE projects, VLC, and GIMP (even though some companies pay their employees to contribute to them), but includes for-profit company and funded public-sector organization projects that accept volunteer contributions, such as Google’s Trillian, Apple’s Swift, CERN’s ALICE, and Microsoft’s Typescript. Our aim is to minimize the number of false positives in the dataset, but we are not interested in the number of false negatives. We do not aspire to create a comprehensive dataset of enterprise projects, but one that contains a sufficient number to conduct generalizable empirical studies.

2 CONSTRUCTION AND EVALUATION

An overview of the dataset’s construction process is depicted in an extended version of this paper [45]. The projects were selected from GitHub by analyzing the GHTorrent [17, 18] dataset (release 2019-06-01) by means of the simple-rolap relational online analytical processing and rdbunit relational unit testing frameworks [19].
Following published recommendations the code and primary data associated with this endeavor are openly available online.\(^1\)

The basic premise for constructing the dataset is that an enterprise’s employees are likely to contribute to a project developed by their organization using the email account provided by it. Furthermore, it is unlikely that pure volunteer projects will have contributors using emails from a single enterprise-related domain address. Based on this premise, we identified projects where a large number of commits were contributed through accounts linked to the same enterprise email domain address. To increase the dataset’s quality we then removed project clones [46], and only retained projects having more than the identified dataset’s average stars (14) and commits (29). Finally, we created one table with diverse details regarding each selected project and one with details regarding each associated enterprise domain. The following paragraphs detail each step, starting from the creation of two tables: valid enterprise domains and probable company domains.\(^2\)

Valid enterprise domains. This table was created by filtering all email domains found in the users’ email table (Table domains). We did this by examining frequently occurring email domains, and creating rules to retain only those associated with enterprise development. Specifically, we removed from the set of domains a blacklist (Table domain blacklist) containing those associated with: email providers; top and second level organization domains, and thereby the many associated with volunteer open source organizations; open source hubs; top and second level educational domains and, explicitly, the domains of more than 20 hand-picked universities; individuals. We did not remove government organizations and research centers as these mainly operate as enterprises with professional developers. When in doubt, we looked up company emails in the RocketReach provider of company email format details.

Probable company domains. This table was created by identifying domains that are likely to belong to companies from publicly available data and domain heuristics. We obtained the domains associated with large companies in two ways. First, we screen-scraped, downloaded, and filtered the data associated with the Fortune Global 500 companies: the largest corporations across the globe measured by revenue (Table fortune global 500). Second, we obtained the US Securities and Exchange Commission (SEC) yearly company filings that are made in machine readable form and extracted from them the company domains. Specifically, we obtained from EDGAR—the SEC’s Electronic Data Gathering, Analysis, and Retrieval system—the XBRL files associated with two forms, namely a) Form 10-K, that gives a comprehensive summary of a company’s financial performance (Table sec 10 K domains), and b) Form 20-F, that provides an annual report filing for foreign private issuers—non-U.S. and non-Canadian companies that have securities trading in the U.S. (Table sec 20 F domains).

We then extracted the internet domain (e.g. intel.com) associated with each company from the XBRL files. We obtained the company domains by looking at the XML name space used in the files, which in most cases contains the company’s domain. We combined the three sources into the Table distinct company domains and complemented it with the Table valid enterprise domains filtered to include only records associated with top and second level commercial domains such as .com, .co.uk, .com.au.

From enterprise organizations to their projects. As a next step we combined the two tables with another listing domains registered for GitHub organizations (Table org domains), to get tables with user domains linked to GitHub organizations—Tables valid enterprise users and probable company users. The intuition here is that many companies developing software on GitHub will have configured a company organization under their domain name. Combining the two tables with the GHTorrent Projects table yielded the corresponding projects hosted under a GitHub organization: valid enterprise projects and probable company projects. These two tables were then linked with a table of each user’s email domain (Table user domain) and one identifying each commit’s committer (Table project commit committer domain), giving the number of committers in each project associated with the corresponding organization: valid enterprise domain committers and probable company domain committers. This stage ended by selecting projects from organizations having a minimum number of committers appearing on GitHub with an email associated with the organization’s domain giving the tables multi committer valid enterprise projects and multi committer probable company projects. The employed floor values (ten and five correspondingly) were selected to exclude projects associated with individuals operating under a personal but commercial-looking domain (e.g. johnsmith.com).

Enterprise-dominated projects. To cover enterprises that may not have GitHub organizations registered with emails under their domain, we also established in each project a rank of committers with valid enterprise email addresses according to their number of commits (Table project commit domain rank), and obtained those projects having committers from the same organizations as the topmost three (Table same domain top committers).

Final filtering and reporting. For the three types of possible enterprise projects we then formed their union (Table candidate projects), combined their metrics (Table merged projects), removed duplicate projects (Table deduplicated projects), combined records referring to the same project (Table merged domain projects), and joined them with the number of their commits (Table project commit count) and their stars (Table project stars), to select those with above average such metrics (Table above average projects). For each one of the shortlisted projects, we git-cloned from GitHub the project’s repository and calculated its basic size metrics in terms of files and text lines (Table size metrics). (Due to churn from the date the GHTorrent dataset was published, not all repositories could be retrieved for measuring project size.)

Finally, to provide context for each project, we combined this table with each project’s earliest and most recent commit (Table commit range), number of commits (Table project commit domain count) and committers (Table project commit domain) for each committer domain, number of commits (Table project commit author domain count) and committers (Table project author domains) for each author domain, total number of committers (Table project

\(^1\)https://doi.org/10.5281/zenodo.3742973

\(^2\)In the interest of readability, this text replaces the underscores in the table names with spaces.
committer count) and authors (Table project author count), size metrics (Table project size metrics), project license as provided by the GitHub API (Table licenses), as well as details about the derivation of the corresponding domain. This process created the table enterprise project details and the corresponding report enterprise projects.

We manually evaluated a random sample of an earlier version of this dataset, following the systematic review guidelines by Brereton et al. [6]. The sample size was calculated at around 378 using Cohen’s sample size and correction formulas [7] (95% confidence, 5% precision). To keep the raters we complemented the sample with 22 GitHub projects randomly selected from a set of projects with similar quality characteristics that were part of the dataset (Table cohort projects). The third and fourth authors were instructed to individually label the 400 projects as enterprise or not based on the definition in Section 1. To improve the labeling’s reliability the two raters did not know the employed heuristics, and were also asked to complete the main reason the project was open source and write a few words to support their decision. Their ratings led to 78% inter-rater agreement and 29% reliability using Cohen’s kappa statistic. The second author then resolved the conflicts by majority vote; after excluding the 22 irrelevant projects, 89% of the 378 projects were finally identified as enterprise. We used the bootstrap method [11] with 1000 iterations to establish a confidence interval (CI) for the percentage of enterprise projects in our sample; the 95% CI was calculated at [87–93]%. To generalize, we employed the dataset’s precision, our method was ated with projects are: com (13 494 projects), io (763), de (383), gov (339), net (256), ru (142), fr (134), cn (120), br (118), and uk (111). Similarly, out of 5 097 owners, those associated with the highest number of GitHub projects are: Microsoft (855 projects), GoogleCloudPlatform (123), twitter (93), 18F (90), udacity (82), SAP (79), Netflix (79), hasicorp (77), and GoogleCloudPlatform (77).

In very few projects does development appear to be exclusively controlled by the enterprise: we found 90 projects (0.5%) where all commits came from an enterprise committer and 220 projects (1.3%) where all commits came from an enterprise author. We were expecting these projects to be small, but in fact they sport an average line count of 453k for projects with exclusively enterprise authors and 976k for projects with exclusively enterprise committers. Considerable development seems to happen through pull requests, with 95% of the projects having pull requests associated with them, with an average of 161 pull requests per project.

In total, according to their SPDX identifiers, the projects are licensed using 29 different open source licenses. The two most common licenses used are the MIT (4 340 projects) and Apache 2.0 (3 761 projects), with the GPL version 2 or 3 license used only by 780 projects. This finding indicates that few enterprise open source projects seem to follow a business model based on relicensing GPL code for proprietary development. Surprisingly, for 3 535 projects no license was found, and for 3 374 projects the license did not match one with an SPDX identifier.

We compared the earlier version of this dataset mentioned in Section 2 against the Reaper dataset of engineered software projects [36] in terms of stars, commits, pull requests (PRs), authors, and commit-
| Metric | Min E | Max (k) E | Avg E | Stddev E | Min R | Max (k) R | Avg R | Stddev R |
|--------|-------|------------|-------|----------|-------|------------|-------|----------|
| Stars  | 15    | 80         | 51    | 355      | 11    | 1661       | 221   |          |
| Commits| 30    | 304        | 383   | 1159     | 70    | 5323       | 1196  |          |
| PRs    | 0     | 25         | 42    | 161      | 3     | 672        | 94    |          |
| Authors| 1     | 26         | 5     | 27       | 2     | 213        | 10    |          |
| Committers| 1 | 26 | 5 | 22 | 2 | 208 | 7 |          |

form login-name/project-name, from which 1,849,500 were successfully associated with a project ID of GitHub. Null values were substituted with zero in both datasets, thus metrics were calculated on the basis of the entire dataset sizes (17,252 for this, 1,849,500 for the Reaper). It appears that in all dimensions this dataset is considerably richer than the Reaper one. The difference most likely stems from this dataset’s considerable selectivity, as it contains two orders of magnitude fewer projects than Reaper.

4 RELATED WORK

While the relationship between academic or semi-academic institutions and open source software has been favorable [29], with large open source projects such as the Berkeley Software Distribution (BSD) [39] originating from them, this has not always been the case for business. The relationship between business and open source software was often tense in the past, with GPL-licensed software described as “an intellectual property destroyer”, un-American, and “a cancer” [34]. Meanwhile, others asserted that open source was compatible with business [22], and researchers quickly identified several business models that are based on open source software [1, 4], as well as significant industrial adoption of open source software products [44]. In short, research associated with the involvement of enterprises in open source software can be divided into four areas [27]: a) company participation in open source development communities [5, 23]; b) business models with open source in commercial organizations [4, 20]; c) open source as part of component based software engineering [2, 30]; and d) usage of open source processes within a company [13, 31]. We consider our study part of the first area. According to Bonacorsi et al. [5], companies participated in one of the most active projects on SourceForge as project coordinators, collaborators in code development, or code providers. Hauge et al. [21] also identified the role of component integrator. By providing their proprietary software to the open source community, companies can benefit from reduced development costs, advanced performance, repositioning in the market, and additional profit from new services [27]. Still, the provided software should be accompanied by adequate documentation and information to help the community members engage in it [21]. Although companies marginally participated in open source projects in the past, the participation has recently increased, especially in the larger and more active projects, with a crucial part of the open source code being provided by commercial organizations, particularly small and medium-sized enterprises (SMEs) [32]. For instance, 6%–7% of the code in the Debian GNU/Linux distribution over the period 1998–2004 was contributed by corporations [41].

Similarly, German and Mockus [14] linked identical contributors of CVS repositories with multiple names or emails of different spelling. Using their infrastructure they identified the top contributors of the Ximian Evolution project, and found that the top ten contributors were Ximian employees and consultants, and also that private companies such as RedHat, Ximian and Eazel, severely affected the development of the GNOME project [15], similarly to the way the Mozilla project was mainly developed by Netscape employees [35].

5 RESEARCH IDEAS

The provided dataset can be employed in various ways. First, it can be used to study the involvement of enterprises in OSS development by examining whether they are mostly takers or givers, their roles within projects, and how they shape a project’s evolution and success [5]. Second, it can be employed in studies regarding OSS business models, to investigate how their choice is affected by different enterprise characteristics such as the employees’ education level, the enterprise’s age, size, service variety, and whether it is family-owned or not [20]. Third, it can be used for research on the composition and structure of OSS supply chains and value chains, particularly to identify the added, deleted, and unchanged dependencies and their effect between releases for different types of packages such as build and test tools [10]. Furthermore, it can be employed in studies concerning enterprise-driven global software development, to measure benefits and tackle issues induced from the physical separation among project members such as strategic, cultural, communication, and knowledge management issues [25]. Another use involves identifying product or process differences between enterprise and volunteer-driven software development in terms of cost, service and support, innovation, security, usability, standards, availability, transparency, and reliability [37]. Finally, it can be used to study enterprise regulatory, compliance, and supply chain risks, to investigate the risk domains that enterprises face when engaging in OSS development, the available sources of risk mitigation, and the heuristics by which managers apply this understanding to manage such projects. From these insights, formalized risk mitigation instruments and project management processes can be developed [16].

ACKNOWLEDGMENTS

This work has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 825328.

REFERENCES

[1] Stephanos Androusetis-Thotokis, Diomidis Spinellis, Maria Kechagia, and Georgios Gousios. 2011. Open Source Software: A Survey from 10,000 Feet. Foundations and Trends in Technology, Information and Operations Management 4, 3–4 (2011), 187–347. https://doi.org/10.1561/1000000026.
[2] Claudia Ayala, Øyvind Hauge, Reidar Conradi, Xavier Franch, Jingyue Li, and Ketil Velle. 2009. Challenges of the Open Source Component Marketplace in the Industry. Vol. 299. 213–224. https://doi.org/10.1007/978-3-642-02032-2_19.
[3] Adrian Bachmann and Abraham Bernstein. 2009. Software Process Data Quality and Characteristics: A Historical View on Open and Closed Source Projects. In Proceedings of the Joint International and Annual ERCIM Workshops on Principles of Software Evolution (IWSE) and Software Evolution (Evol) Workshops (IWSE-Evol ’09). Association for Computing Machinery; New York, NY, USA, 119–128. https://doi.org/10.1145/1595808.1595830.
