Diversity in Open Source Community and its Impact on Code Quality and Productivity

Xiaozhe Yao[1], Yingying Chen[1], and Weijie Niu[1]

[1]University of Zurich

November 25, 2019

Abstract

In 2018, Github has restricted several developers' accounts from Iran and other countries due to the impact of U.S. trade restrictions. As a result, the concern for the diversity in Open Source Community arises. Some people argued that Github has to find some middle ground between corporate puritanism and the diverse culture that surrounds it("GitHub threatens to shut down a repository for using the word 'retard' — Hacker News", n.d.). Diversity may mean a lot for social activists, politicians, etc. But as software engineers, it is still in doubt whether the diversity will have impacts on productivity and quality. We will explore the relationship between the project activity/productivity and diversity of the project developer community.

With this work, a new software engineering perspective on the impact of the diverse culture of open source community will arise.

Research Questions

We have three key research questions:

1. Diversity, as a quality, is hard to measure. How do we quantify and measure the diversity of open source projects?
2. Is there any relation between the code quality and diversity of OSS? To answer this, we need to evaluate the quality of source code.
3. Is there any relation between the productivity and diversity of OSS? To answer this, we need to evaluate the productivity of open source projects.

Related Work

Since the open-source community is more open, geographically distributed, it can accommodate more possibility and is famous for diverse culture. In our project, we want to figure out how diversity influence code quality and productivity. First, we pick up our indices for diversity. According to Harrison and Klein (Harrison & Klein, 2007), they defined diversity as the distribution of differences among the members of a unit with respect to a common attribute and classified diversity to three types, separation, variety, and disparity respectively. Thus, we organise our indices in the following three aspects. Separation, variety, and disparity are respectively understood as differences in attitude or position, differences in categorical characteristics, and differences in power or status hierarchy (Solanas, Selvam, Navarro, & Leiva, 2012).

1. Separation diversity. The impact of separation diversity is conceptualized as culture (Daniel, Agarwal, & Stewart, 2013), so we select the workplace(time zone), the hometown and company of the participant to estimate for cultural separation diversity.

2. Variety diversity. It can be reflected in dispersion in project participant roles(Daniel, Agarwal, & Stewart, 2013) which are categorized into developers, administrators and active users. Developers are those who make CVS commits(Daniel, Agarwal, & Stewart, 2013) while administrators take charge of release or the owners of repositories. Active users are identified as individuals who reported bugs(come up with issues), requested features or support, or participated in discussion forums. As for another categorical characteristic, gender could be a relative factor in OSS project. Gender is proven to be positive and significant predictors of productivity(Daniel, Agarwal, & Stewart, 2013). However, we have not known how

gender relates to code quality. Thus, we intend to figure out If gender proportion adversely or positively affects project's code quality and how significant this effect is.

3. Disparity diversity. It reflects variation in participants' contribution-based reputation (Daniel, Agarwal, & Stewart, 2013). The level of one participant's contribution is measured as the sum of accepted commits and all responses to issue and corresponding comments and likes.

For our datasets, we cannot get access to the hometown and gender of participants but we can infer hometown from their usernames and infer gender on genderComputer(Vasilescu, Capiluppi, & Serebrenik, 2013).

Method

Data Collection

Prior to 2018, there are over 100 million repositories and 31 million developers on GitHub. It would be tedious and costly to analyse all of them, thus we conduct preliminary research to find the repositories and developers that could address the diversity. We found two important facts about Github:

1. Most of them are small-size projects and has little attention. (i.e. only 1 contributor and less than 1000 stars).
2. For repositories with more than 1 contributors, most of them have only a few active contributors. Most contributors only contribute several lines of code once.

In order to reduce the cost of analysis, we use the following criteria to find qualified repositories and contributors.

1. For contributors, they will be counted if they have ***at least 1 code/issue submission*** once a week.
2. For repositories, they must have ***more than 5 active contributors.***

For those repositories, we will record the properties such as (repository_name, owner_name, stargazer_count, creation_date, main_programming_language, creation_date). For contributors, we will record the properties as (name, location).

After the record of primary properties of repositories, we will then find proper repositories for code quality analysis. Since different languages are usually equipped with different static analysis toolkits and include different aspects of quality issues (e.g. lowercase class names is not an issue in Golang since it represents private class, but in other conventions, it is a style issue because people usually use uppercase class names), use uppercase class names), it makes no sense to compare the quality among different languages. Therefore we decided to focus on the repositories whose main_programming_language are the same. Since the topic of this research is on the diversity and team work, a programming language which is designed for large, cooperative teams of programmers. With this standard, as described by Google in ("Go at Google: Language Design in the Service of Software Engineering", n.d.), Golang works at scale, for large programs with large teams of programmers working on them. Besides of this character, Golang is also the TIOBE language of the year in 2016 and has attracted a huge interests. For example, lots of successful large-scale softwares are written in Golang, such as Kubernetes, CloudFlare and Dropbox.

We will then download the source code and its commits history of golang repositories for a further analysis of code quality.

Pre-Processing

Inferring Gender

There are already lots of work on the inference of gender from name. We adopt the method described in (Vasilescu, Capiluppi, & Serebrenik, 2013), which combines a number of transformations, diminutive resolution and heuristics, and the reported precision is 93%. It should be fine for this task and can be better utilized with some data augmentation (e.g. add some common names for male/female for some countries with a lot of developers).

Quantifying Diversity

There are several different diversity indices, such as Blau's Index for variety as described in (Blau, 1977), Standard deviation for separation and coefficient of variation for disparity. We will use these techniques to quantify diversity.

Code Quality Measurement

Thanks to the standard toolkit provided by the golang community, we are able to analyse the code quality by open source software named GoReportCard, as seen in (Schaaf & Smith, 2015–). The toolkit integrate several measures, including gofmt (find unformatted code), govet (examines suspicious constructs), golint(examines if comments exists and are in the proper format, and other linting functions), gocyclo(calculates cyclomatic complexities of functions, warns for functions with cyclomatic complexity > 15), ineffassign (detects ineffectual assignments), misspell (find commonly misspelled English words), etc. All these aspects of quality will be taken into consideration in our analysis, since they cover different aspects and according to the Golang user manual, they are all quality issues defined by the whole community.

We may also use bug tracking systems, such as issue history, findbug("FindBugs™ - Find Bugs in Java Programs", n.d.) to evaluate the code quality issues from the aspects of code vulnerability.

Productivity Measurement

Within the limited timeframe, only objective productivity measurement will be used in this section. We will start from three aspects, i.e. lines of code addition/deletion, issues lifecycle (more precisely, the average time when issues open to when they are closed/solved), code quality improvements etc.

Besides these, a subjective survey could be conducted in online forums, such as reddit etc. We will then ask the developers Github name, affiliation, attended projects, their perceived productivity.

Regression Analysis

We will conduct regression analysis to examine the relations between diversity and code quality/productivity. These include two steps:

1. Single Variable Analysis where independent variables are location/timezone/gender etc.

2. Combined Analysis where independent variables are the variety, separation, or disparity.

Project Lifespan Analysis

In prior research, we have found an interesting fact that lots of projects, even started by big companies, dead or not maintained for a long time, see Fig.1 for the exact figures. It is a huge lost for the community to abandon a project. To help with the situation, we decided to conduct an analysis on the project lifespan with the diversity. With this analysis, we hope we could provide some practical advice for the project maintainer to make the project's life longer.

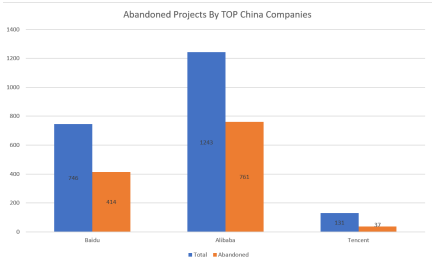

Figure 1: Abandoned projects count by top Chinese Companies

Post-Processing

Finally, with all these analysis, we will use data visualization to demonstrate our research results.

Conclusion

In this research, we aims to explore the relations between diversity and the quality/productivity of open source projects. We use regression analysis to determine the degree to which our definition of diversity are influencing the quality and productivity. We hope this research could help OSS maintainers improve their project lifespan and other key factors.

References

. https://news.ycombinator.com/item?id=9966118. Retrieved from https://news.ycombinator.com/item?id=9966118

. https://talks.golang.org/2012/splash.article. Retrieved from https://talks.golang.org/2012/splash.article

. http://findbugs.sourceforge.net. Retrieved from http://findbugs.sourceforge.net

Blau, P. M. (1977). *Inequality and heterogeneity: A primitive theory of social structure* (Vol. 7). Free Press New York.

Daniel, S., Agarwal, R., & Stewart, K. J. (2013). The Effects of Diversity in Global Distributed Collectives: A Study of Open Source Project Success. *Information Systems Research*, *24*(2), 312–333. https://doi.org/10.1287/isre.1120.0435

Harrison, D. A., & Klein, K. J. (2007). What's the difference? diversity constructs as separation variety, or disparity in organizations. *Academy of Management Review*, *32*(4), 1199–1228. https://doi.org/10.5465/amr.2007.26586096

Schaaf, H., & Smith, S. (2015–). Go Report Card: A report card for your Go application. Retrieved from https://www.goreportcard.com/

Solanas, A., Selvam, R. M., Navarro, J., & Leiva, D. (2012). Some Common Indices of Group Diversity: Upper Boundaries. *Psychological Reports*, *111*(3), 777–796. https://doi.org/10.2466/01.09.21.pr0.111.6.777-796

Vasilescu, B., Capiluppi, A., & Serebrenik, A. (2013). Gender Representation and Online Participation: A Quantitative Study. *Interacting with Computers*, *26*(5), 488–511. https://doi.org/10.1093/iwc/iwt047