Case study: Relation between number of stars and features of software engineering in GitHub repositories

Emanuel Mazzer
Universidade Tecnológica Federal do Paraná
VIA ROSALINA MARIA DOS SANTOS, 1233
Campo Mourão, Brasil
emanuelgiroldo@gmail.com

João Vitor Foralosso Gris
Universidade Tecnológica Federal do Paraná
VIA ROSALINA MARIA DOS SANTOS, 1233
Campo Mourão, Brasil
joaogrisxv@gmail.com

ABSTRACT

This paper reports a study on the relationship between GitHub software repositories, considering the repositories that contain traces of software engineering. We analyzed the 140 repositories with the highest number of stars. The analysis was done in order to find the common characteristics among the repositories and relates them to the number of stars that such a repository has, so that it is possible to find the main characteristics that make the repository popular.

Keywords

GitHub; Mining; Repositories; Software Engineering

1. INTRODUCTION

With the great popularization of code-hosting platforms with version control, such as GitHub, the search for these services has created a new community. These platforms are widely used by programmers to publicize their work and to collaborate with other community programmers. In this way, several repositories began to become popular and attract the attention of a large number of people and enabling them to collaborate. The popularity of repositories is a very important factor, it is from this popularity that administrators can find out if their repository is attracting new users, or if their software is being accepted [2].

This paper reports a study on the relationship between GitHub software repositories, considering the repositories that contain traces of software engineering. We analyzed the 140 repositories with the highest number of stars. The analysis was done in order to find the common characteristics among the repositories and relates them to the number of stars that such a repository has, so that it is possible to find the main characteristics that make the repository popular[3].

The results were satisfactory, showing that a good part of the analyzed repositories use the same programming language. In addition, it has been discovered that the number of commits and the tags of the repositories. influence its popularization.

© 2017 ACM. ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

2. TEXT ORGANIZATION

The sections of the text are arranged as follows: **Related Work** section, where we explain the work that contributed to this study, both the datasets and the article, **Methodology**, section where we explain how we executed the study, **Results**, section where we show and we explain the results obtained, **Conclusion**, it's the final section where we show what we concluded on our study.

3. RELATED WORK

For the analysis, it was necessary to search for the repositories that contained the largest number of stars, after finding them it was necessary to find the repositories that contained traces of software engineering. The two datasets were obtained from other studies already performed.

3.1 Top starred repository dataset

For this set, GitHub.com's open-source projects were crawled and a list of projects that contained the most stars was drawn. In GitHub, a user can choose to star to a repository, and that should represent that he likes the project. For each project the following characteristics were obtained:

- **Username** The username on which the project resided.
- **Repositorio.Name**: Repository name.
- **Description**: Description of the repository
- **Last.Update.Data**: The last time the repository was updated.
- **Language**: The project language.
- **Nomber.of.Stars**: The number of stars that the repository has.
- **Tags**: All tags in the repository.
- **Url**: Project URL.

This dataset was created by the american Chris Willden, and was obtained through Kaggle [1], which is a platform for competitions in which statisticians and data miners compete to produce the best models to predict and describe datasets. The database was last updated June 24, 2017 and this version was used in this study.

3.2 Dataset of repositories with traces of software engineering

For this set, GitHub.com's open source projects were crawled and a list of over one million eight hundred thousand repositories was extracted. This set was obtained with the intention of testing a framework called Reaper that was developed by a group of teachers together with some students mainly at the Rochester Institute of Technology. The framework aims to analyze software repositories and indicate those that contain traces of software engineering [4]. The classification was made using several characteristics, the main ones being:

- **Community**, such as community contriubution data. Since the presence of a community of developers indicates that there is collaboration and cooperation involved in the development of the software system, which is partial evidence that the repository contains software engineering.
- **Continuous integration**, as quality data, since the practice of continuous integration ensures that the software system of this repository is in constant evolution and is stable for development. The use of Continuous Integration is further evidence that software design may have traits of software engineering.
- **Documentation**, as maintenance data, because the documentation should help in understanding the software system for maintenance purposes, this indicates that the author thought about maintenance, which shows that the repository contains traces of software engineering.
- **History**, as data of sustained evolution, that is, software receives corrections and changes with a certain frequency, this presence of changes and improvements indicates that the software system contains traces of software engineering.
- **Issues**, such as repository management data, such as requirements management, scheduling, tasks, defects, and software releases. Thus, evidence of the use of project management tools may indicate traces of software engineering.
- **License**, including a license in the repository explicitly dictates the rights, or lack thereof, of the user making copies of the repository. The presence of a software license is required, but alone does not identify whether the repository contains traces of software engineering.
- **Unit testing**, as quality data, since software engineering products need to guarantee product performance, and this guarantee is provided by rigorous testing. Evidence of testing in a software project may indicate traces of software engineering in the project.

This dataset was obtained through the article describing the operation and also the theoretical basis of the framework, there also explained all calculations used for the computation of the characteristics.

4. METHODOLOGY

This chapter will explain all the methodology used to obtain the characteristics analyzed. In addition, it will be commented on the pre-processing performed on the two datasets used in this work.

4.0.1 Pre-processing

In order to facilitate the extraction of the characteristics, the datasets underwent a pre-processing, in which some lines were removed and others added, the modifications made were:

1. In the dataset containing the largest number of stars, the column "number of stars" containing string data was converted to floating, with the possibility of performing calculations with those values.
2. In the data set with software engineering traces, all rows containing null values for the "stars" field were first removed.
3. After that, we selected the 580 repositories with the largest number of stars in the dataset.
4. Then, it was verified which of the 580 repositories were present in the two datasets.
5. In these repositories, the information about the tags that the repositories possessed was added.
6. Finally, repositories that did not contain tags in their descriptions were excluded.

After these steps, there were a total of 140 repositories, containing all the necessary characteristics for the study proposed in this work.

4.0.2 Method

To analyze the data from the 140 repositories that were filtered and preprocessed, we tried to highlight common points among the repositories, what were common practices, and to try to find out how software engineering projects are generally executed. We used the standard deviation of the characteristics to find out which of them were most similar among the repositories. In addition, we performed a related analysis in the tags field, where we tried to find the most used ones in these repositories, in addition we looked for to relate the number of stars and the repositories using the number of stars of each language, ie the number of stars of the repositories that contained that language in the source code of their projects.

In this section we will show the views that were obtained and also the data obtained for comparison of the characteristics.

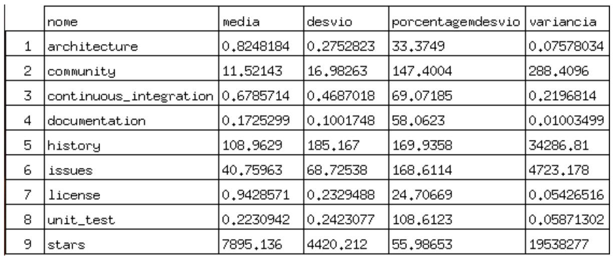
4.1 Standard deviation e variance

We used standard deviation and variance to help us choose the features that were most similar across repositories. For this we use only the characteristics with numerical values. In addition to the standard deviation, we also use a mean to know the ratio of standard deviation to standard deviation.

With this data we can know which features vary less between the repositories, to improve our choice of the most common characteristics we did a table that shows this data, this table is represented by Figure 1.

4.2 WordCloud with the Tags

In a view such as WordCloud, each word has its size governed by relevance in the text. In this case it is a simple counting of occurrences of a certain word in the "tags" field. A word quoted more often will have a proportionally larger size than a quoted word less often.



	nome	media	desvio	porcentagemdesvio	variancia
1	architecture	0.8248184	0.2752823	33.3749	0.07578034
2	community	11.52143	16.98263	147.4004	288.4096
3	continuous_integration	0.6785714	0.4687018	69.07185	0.2196814
4	documentation	0.1725299	0.1001748	58.0623	0.01003499
5	history	108.9629	185.167	169.9358	34286.81
6	issues	40.75963	68.72538	168.6114	4723.178
7	license	0.9428571	0.2329488	24.70669	0.05426516
8	unit_test	0.2230942	0.2423077	108.6123	0.05871302
9	stars	7895.136	4420.212	55.98653	19538277

Figure 1: Graph of Relation between Fields

The figure 2 shows a word cloud created from the "tags" that were contained in the dataset.

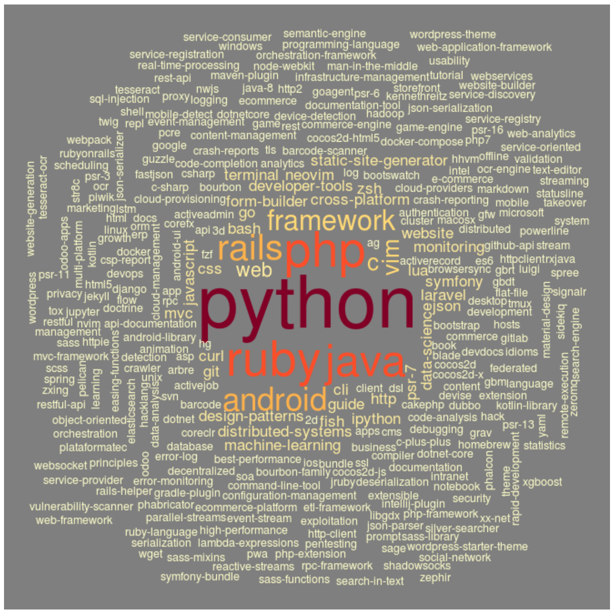


Figure 2: WordCloud das Tags

Observing the wordcloud it can be noticed that only a small group of tags stands out, while the great majority has few repetitions, leaving evident the most common among the repositories.

4.3 Graphic of the use of languages by number of stars

In order to find out which languages were most common among repositories, we tried to show which ones appeared and also how much each language influences the number of collaborators. In the figure 3, the languages are ordered by the sum of the number of stars of the repositories containing that language.

5. RESULTS

The result of our work was obtained and synthesized by analyzing the data of the sets, as well as analyzing the visualizations and patterns found. In order to achieve the objective of showing the relationship between the projects that contained traces of software engineering and their popularity (measured in number of stars), some characteristics that were common among them were highlighted.

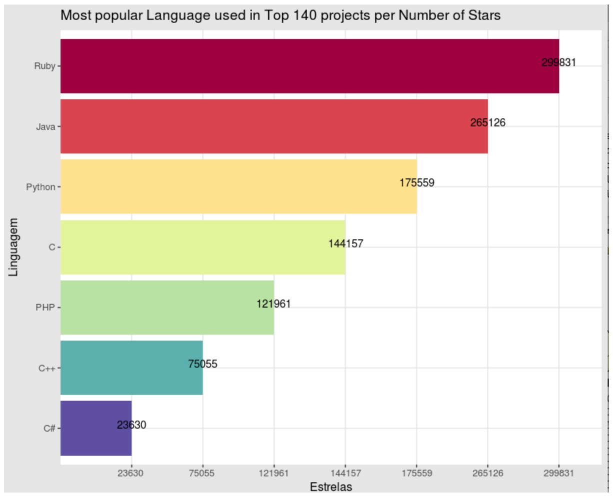


Figure 3: Graphic of the use of languages by number of stars

5.1 Analysis of the standard deviation table and variance

Making a visual analysis of the results obtained on the deviation of each of the characteristics with numerical values of the repositories we were able to extract the information that the repositories varied little. We observed that the values varied widely, and that only two characteristics presented little variation:

- **Architechture**, which shows that most of the most popular projects contain a similar level of organization of architecture, that is, in the majority, the projects are done thinking about maintenance and ease of evolution. These architectural traits show that projects implement some design pattern.

- **License**, including a license in the repository explicitly dictates the rights, or lack thereof, of the user making copies of the repository. The characteristic is not determinative to say whether or not a project contains software engineering.

5.2 Analysis of WordCloud of tags

n an analysis, also visual on the WordCloud of the tags of the Figure 2 tags, we extract the most common tags based on size. The top tags were, in descending order:

1. **Python**
2. **php**
3. **Ruby**
4. **Java**
5. **Rails**
6. **Android**
7. **Framework**

5.3 Chart Analysis of language usage by star number s

By visually analyzing the graph of the use of the languages of Figure 3, we were able to show that the projects that received the most stars contained codes in the following languages, in descending order of number of stars:

1. **Ruby**
2. **Java**
3. **Python**
4. **C**
5. **php**
6. **C++**
7. **C#**

6. CONCLUSION

At the end of this work, we conclude that some features are common among repositories such as **Architechture** and **License**, but these characteristics are not decisive enough to reveal a pattern. Thus, in order to obtain a more precise result, besides these characteristics, other facts should be studied more deeply, such as the experience of the collaborators, the time of collaboration and also information about the customs of the community that is behind that project.

The analysis of the tags showed a good pattern, evidencing the most common among the repositories, but this could have a slightly better result if the repositories were analyzed separately according to their predominant language.

The analysis about the languages of the repositories with more stars showed us that some languages are more used, but can not guarantee that this factor is determinant in the process of popularization of the project.

Thus, we can conclude that there is a relationship between the characteristics of the projects that contain traces of software engineering, but for them to be better evidenced, a deeper study in the repositories is necessary. The characteristics of the data sets used in this work indicate that a pattern exists but can not make clear where and what the pattern is.

in your paper.

7. REFERENCES

[1] Kaggle: Your home for data science. https://www.kaggle.com/, 2017.

[2] H. Borges, A. Hora, and M. T. Valente. Predicting the popularity of github repositories. In *Proceedings of the The 12th International Conference on Predictive Models and Data Analytics in Software Engineering*, PROMISE 2016, pages 9:1–9:10, New York, NY, USA, 2016. ACM.

[3] H. Borges, A. Hora, and M. T. Valente. Understanding the factors that impact the popularity of github repositories. In *2016 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 334–344, Oct 2016.

[4] N. Munaiah, S. Kroh, C. Cabrey, and M. Nagappan. Curating github for engineered software projects. *Empirical Software Engineering*, 22(6):3219–3253, 1993.