Evaluation Model on the Usage of Java Project Deployment Tool in Open Source Community

Chengrong Yang
School of Software
Yunnan University
Kunming, China
Moreyang@outlook.com

Hao Li*
School of Software
Yunnan University
Kunming, China
lihao707@ynu.edu.cn

Yan Kang
School of Software
Yunnan University
Kunming, China
Kangyan@ynu.edu.cn

Chenyang Lu
School of Software
Yunnan University
Kunming, China
luchenyanglcy@163.com

Junsong Liu
School of Software
Yunnan University
Kunming, China
liujs1995@live.com

***Abstract*—Building and managing projects is an integral part of the software life cycle. The use of management tools will affect the cost and efficiency of development. Maven, Ant, and Gradle are specialized tools for building and managing java-related projects. This paper analyzes the efficiency of these tools. In particular, getting information about Java open source projects which built with the project deployment tools Maven, Ant, and Gradle in the GitHub open source community through crawlers. It analyzes the impact of using tools on project development and quality by comparing Java open source projects that use deployment tools with those do not. It tested the following characteristics: average commit velocity, number of bug-referencing commits, number of issues recorded, usage of continuous integration, number of pull requests, and distribution of commits per author, designing evaluation factors such as project influence degree, project activity degree and project popularity degree. At the same time, this paper improved PageRank algorithm to measure the project. It is found that Java projects using Maven tools were relatively rare. In addition, there were very few significant differences in any of the metrics we used to compare Maven-used and Maven-unused projects. Therefore, our results do not reveal any observable benefits from using Maven.**

Keywords—management tool, evaluation factors, GitHub open source community, PageRank algorithm, evaluation model

I. Introduction

In recent years, with the deep combination of software collaborative development technology and social networks, the social development paradigm has occupied an indispensable position in the current software development process [1]. As of April 2018, the 15.777% Java programming language ranked first in the TIOBE programming languages, an increased of 0.21% compared to the April 2017 growth rate. At the same time, a great number of project developers who develop Java projects will choose project construction tools in the Java project development process [2]. According to the Dzone websites' statistics, at present these three project development tools that are Apache Maven [3], Apache Ant [4] and Gradle are widely used. Commits, pull requests, watch, fork, and star based on the mechanisms of provided in the GitHub open source community, combining with the pom.xml, build.xml and build.gradle configuration file attributes in the Java project development tools, the evaluation factor is designed to measure the project importance.

In order to measure the influence of project construction tools in the open source community on project development, this paper proposes an evaluation model for the use of Java project construction tools in the open source community. This model measures the importance of the Java open source projects in three dimensions which are project impact, project activity and project popularity, then evaluates those with the PageRank algorithm. Specifically, The main contribution of this paper is that we designed evaluation factor to measure the project from multiple dimensions, and proposes a project importance evaluation model. The significance of this research includes the following three aspects.

- Analyze the impact of the project build tool on the project and whether project build tool has an impact on project development by comparing quantitative assessment of the project using the project build tool and without using the project build tool.
- Assist the open source to analysis the project importance in quantitative, and project developers can get references through the quantitative evaluation of each Java open source project and the comparison of ranking.
- Guide project development. When a project developer selects a project from a number of open source projects as a basic version for iterative development, it can evaluate the importance of projects, quantitively analyze the project's reliability and importance, and then recommend the high-quality project to the developer.

II. Research Background

The emergence of distributed version control system gave rise to the open source community and promoted the development of open source software. Reference [5] analyzed the combination of social network and software development from the perspective of the GitHub open source community. Reference [6] introduces the Pull-requests code review tool provided in the GitHub open source community. Reference [7] introduced an analytical search platform that can recommend high quality open source software. Reference [8] proposes some influencing factors such as the speed of problem solving and the speed of problem by analyzing the development process of GitHub open source software. Evaluating the various attributes of the open source project in GitHub and evaluating the impact of the project build tool on the project development can provide a reference for the project developer to choose the project build tool. Reference [9] provides a preliminary explanation and analysis of the introduction and application of Maven in the Java project. Reference [10] analyzes and summarizes the specific use of Maven in enterprise software products. The traditional PageRank algorithm can give a global order of importance to web pages on the Internet, but the disadvantage is that the old web page is generally higher than the new web page, unless the new web page is a sub-site of a site. Reference [11] analyzed the development status of PageRank algorithm and proposed an improved algorithm for the subject drift phenomenon. Reference [12] combined time characteristic analysis, language link structure analysis and user behavior to improve the PageRank algorithm based on multiple weighting factors and improve the search quality. In [13], several distributed stochastic schemes for calculating PageRank values are proposed based on the traditional PageRank algorithm, and the multi-agent consistency problem is discussed in detail.

III. EVALUATION MODEL CONSTRUCTION

This section begins with introducing the evaluation factor of the project importance assessment. Then we propose an improved form of PageRank as a method of importance assessment. Finally, the importance assessment model is given.

A. Obtain evaluation factors

This paper measures Java open source projects from three dimensions: project impact, project activity and project popularity: project impact is mainly evaluated from the perspective of dependency between Java open source projects and measures the impact of the project through the penetration and outreach of project. Project activity mainly evaluates the project from the project participants' participation in the GitHub open source community. The frequency of project developers' revisions to a project and the interaction records with others have an impact on the project's activity. At the same time, you can also judge the activity of the project by checking the last interaction record of the project. Project popularity is based on the popularity calculation method of open source projects based on watch number and fork number proposed in [14], adds the number of stars to evaluate the project. The three mechanisms of watch, fork and star can assess project popularity more comprehensive. The specific acquisition methods of each dimension are as follows.

B. Project impact factor

According to the theory of Degree Centrality [15], the more neighbors a node has, the more influence it will have. Based on this theory and depending on the dependencies between projects, the influence of the project is calculated according to the ingress and outbound of the project node. The specific formula is shown in Equation (1).

$$EF(i) = (1-d)\frac{ki_i}{n} + d\frac{ko_i}{n} \tag{1}$$

Where i is the indegree of the project node, ki_i is the outdegree of the project node, ko_i is the ratio of the weights of the dependency and depended relationship n is the number of nodes in the network.

C. Project activity factor

As an open source community for social programming, GitHub provides a variety of social mechanisms for project developers to interact with, one of which is the commit mechanism [16]. Each commit can be regarded as a modified node or correspond to a purpose operation. The number of submissions and the latest time of commit in the project can reflect the project activity.

This paper designs Equation (2), (3) and (4) to calculate project activity.

$$t_i = \frac{\Delta d}{30} month \tag{2}$$

$$T(t_i) = max(0.2, 1-0.2*\frac{t_i}{6}) \tag{3}$$

$$AC(i, t_i) = \alpha * \frac{1}{1+e^{commits(i)}} + \beta * T(t_i) \tag{4}$$

Where t_i represents the recent time interval since the last commit time. $T(t_i)$ is the weight coefficient corresponding to the commit time interval. the above time node is most reasonable to use as the exhaustion cycle by continuously tracking the open source project on GitHub, $AC(i, t_i)$ indicates the activity of the open source project i. $commits(i)$ indicates the number of commits for item i. In order to make the activity metric more accurate, the normalization operation is first performed on the commit number, and then the weight of the project activity is given different weights based on the commit number and the latest update time, and finally linear combination is performed.

D. Project popularity factor

In the GitHub open source community, there is a watch, fork, and star mechanism. If a project developer is interested in an open source project, he can use the watch, fork, and star mechanism to track the project and develop it locally. In this paper, we use these three mechanisms to measure the popularity of the project.

In [14], an open source project popularity calculation method based on watch number and fork number is proposed. In order to quantify the popularity of open source projects more accurately and comprehensively, this paper adds star numbers based on its calculation and designs Equation (5) to calculate the popularity of an open source project:

$$T(t_i) = a * \frac{1}{1+e^{watch(i)}} + b * \frac{1}{1+e^{fork(i)}} + g * \frac{1}{1+e^{star(i)}} \tag{5}$$

Among them, $pp(i)$ represents the popularity of open source project *i*. $watch(i)$, $fork(i)$ and $star(i)$ respectively represent the number of watches, forks, and stars of the open source project *i*.

E. Importance assessment factor

The algorithm used in this paper is an improved one on the basis of PageRank algorithm for web pages sort. However, the PageRank algorithm ignores the user's browsing interest and does not take the key factor of the viewer into account, which leads to the inaccurate of sorting results. This paper solves this problem by introducing a importance assessment factor, which is a linear combination of project impact, project activity, and project popularity. The specific formula is shown in Equation (6).

$$S(i) = \alpha * EF(i) + \beta * AC(i) + \gamma * PP(i) \tag{6}$$

In Equation (6), $S(i)$ denotes the importance assessment factor of open source project *i*. $EF(i)$ denotes the impact index of project *i*. $AC(i)$ denotes the activity of project *i*. $PP(i)$ denotes the prevalence of project *i*.

IV. Improved PageRank's Importance Assessment Algorithm

This paper proposes an improved form of PageRank as an importance evaluation method, which is to train the relationship between feature sets, label the values by the annotated and ranked data set, optimize the parameters between features and establish an importance evaluation model.

Modified PageRank algorithm is to add importance evaluation factors to PageRank algorithm, for the purpose of evaluating the importance of the project more accurately. The formula is designed as Equation (7).

$$PR(i) = \frac{1-d}{C_{total}} + d\sum_{j=1}^{n} \frac{PR(T_j)}{N(T_j)+1} + S(i) \tag{7}$$

Among them, $PR(i)$ represents the importance level of open source project *i*. T_j(j=1, 2,..., n) represents the number of projects on which project *i* depends. *d* rangs from 0 to 1, representings the probability of which a project developer randomly accesses the project; C_{total} represents the number of all project nodes. $N(T_j)$ represents the number of projects on which item T_j depends.

Basing on all the process above, we can construct a flow chart about the open source community Java project with the importance evaluation model. The obtained domain knowledge atlas is shown in Fig. 1.

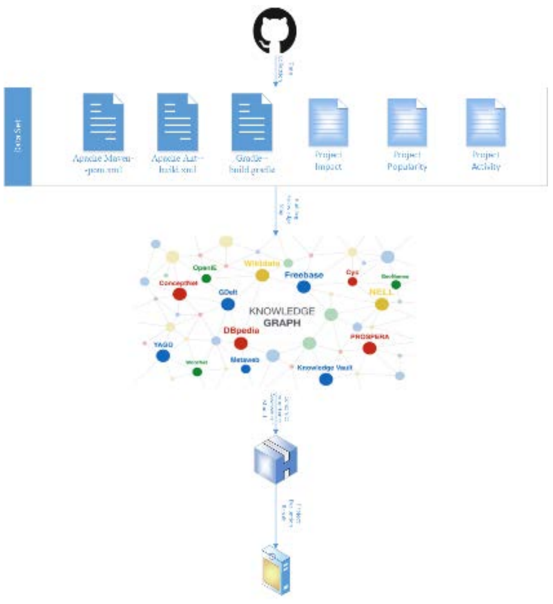

Fig. 1. Project importance assessment model construction flow chart.

V. Experimental Results And Analysis

A. Experimental data set

This article used the crawler [17] and obtained information about the Java open source project in the GitHub open source community as the data set. After removing the data with empty attribute, analyzing and comparing, we evaluate the importance with the model designed in this paper.We simply screened the acquired items, and the projects which have at least one non-empty attribute are selected to be used as the experimental data set. Specifically, 4512 open source projects are selected. Through the cluster analysis of the project description of the selected project, the crawler is used to obtain the open source project related information of the similar project construction tool with similar themes. By selecting simply from existing project, we simply select 3221 programs for experiment dataset that one attribute is not null at least.

B. Choose the parameter

Some of the formula which were put forward in this paper contain two or three of the three parameters, which are α、β and γ. Using the method of Control Variable to adjustment the value of every parameter, calculate the variance of the entire data set, PageRank in the corresponding case, finally we can determine the best parameter according to the value of variance. After adjusting the parameters 300 times，taking the best results: the project activity factor takes α=0.1, β=0.9, the project popularity factors takes α=0.1, β=0.1, γ=0.8, the importance evaluation factors takes α=0.1, β=0.8, γ=0.1.

C. Comparison of project popularity

Fig. 2 is a comparison figure between the calculation method of project popularity proposed in reference[14] and the calculation method of the experimental results proposed in this paper. From Fig. 2, it can reflect the differences between the projects more clearly by the improved method of calculating project popularity by adjusting the parameters.

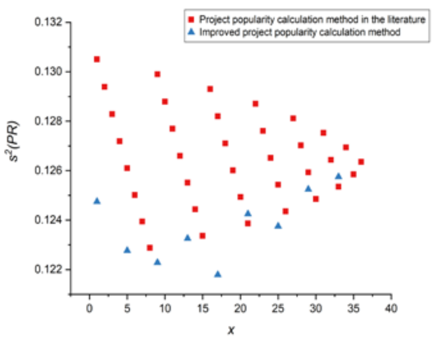

Fig. 2. Project popularity comparison chart.

D. Algorithm Comparison

Different types of nodes in a network have different functions and the importance of each node cannot be given by a single indicator, which leads to the hits algorithm. The comparison between PageRank algorithm and our algorithm is given as Fig. 3 that is a comparison of the experimental results of the traditional PageRank algorithm and PageRank improvements after iteration to steady state.

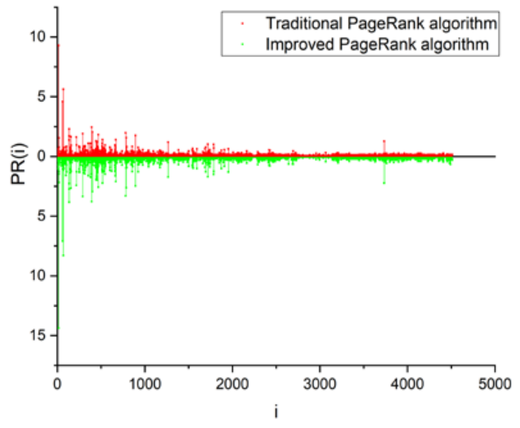

Fig. 3. PageRank algorithm comparison chart.

Table I shows the TOP-5 projects using the traditional PageRank algorithm and the improved form of PageRank to evaluate the importance of the Java open source project.

TABLE I. PAGERANK ALGORITHM TOP-5 PROJECT COMPARISON

Algorithm	**TOP-1**	**TOP-2**	**TOP-3**	**TOP-4**	**TOP-5**
Traditional PageRank algorithm	erlyberly	Activiti	StreamEx	Airsonic	pyx
Improve PageRank algorithm	erlyberly	Activiti	pyx	StreamEx	Airsonic

From Table I, the two algorithms have different evaluation rankings for Java open source projects. From the data of TOP-5, the improved PageRank algorithm has an impact on the project ranking of the project named “pyx”. By comparing the three dimensions of project impact, project activity and project popularity compare the three projects “pyx”, “StreamEx” and “Airsonic”, we know that the “pyx” project is obviously superior to the other two projects, it also proves that the improved form of PageRank is more suitable for the importance assessment of Java open source projects.

Table II shows the variance data of the entire data set PageRank value with the traditional PageRank algorithm and PageRank improved form, it can be seen that the improved PageRank algorithm in this paper can comprehensively reflect the differences between various projects, it is better than the traditional PageRank algorithm, and we can see the specific values in Table I.

TABLE II. PAGERANK ALGORITHM DATA SET VARIANCE COMPARISON

Algorithm	**Data set PageRank value variance**				
Traditional PageRank algorithm	erlyberly	Activiti	StreamEx	Airsonic	pyx
Improve PageRank algorithm	erlyberly	Activiti	pyx	StreamEx	Airsonic

According to the data above we can find that, the degree of impact, activity and popularity of a project has a great impact on its importance assessment, at the same time, there is a dependency relationship between hot spots, the importance evaluation model designed in this paper can measure Java open source projects, and has a good effect.

To demonstrate whether the project building tool is helpful to the development of open source Java projects, selecting the group of items with the highest theme similarity for comparison. As shown in Table III, the data is a group of projects whose theme is "Natural Language Generation". Data comparisons show that the use of a project building tool does not determine the importance of a project.

TABLE III. WHETHER USE PROJECT BUILDING TOOL TOP-5 PROJECT COMPARISON

Project	**TOP-1**	**TOP-2**	**TOP-3**	**TOP-4**	**TOP-5**
Maven-used	0.863770505	0.764199489	0.669487246	0.641412639	0.605614479
Maven-unused	0.859487246	0.731529823	0.670227621	0.665506568	0.620978557

VI. CONCLUSION

This paper proposes a Java project importance assessment model based on evaluation factors, by taking full advantage of the various links between Java open source projects, project and project developers, combining attributes of multiple dimensions, improving PageRank algorithm for importance assessment of open source projects. Concretely, in this paper, evaluation factors are constructed through dependencies between open source projects and related mechanism attributes in the GitHub open source community, measuring the project by combining the three dimensions of Java open source project: project impact, project activity and project popularity. And using the improved PageRank algorithm to establishing an open source project importance assessment model. Finally, use test data sets to evaluate metrics for a given project. The method proposed in this paper according to the multi-dimensional factors affecting project importance assessment, but the use of a project building tool does not determine the importance of a project. The future research will further explore the social connections between the project lifecycle and the project developers, and gradually refine the project importance assessment model.

ACKNOWLEDGMENT

The relevant research in this paper was funded by the National Natural Science Foundation of China (No.61762092 and No.61762089) and the Yunnan Provincial Software Engineering Key Laboratory Open Fund Project (No.2017SE204).

REFERENCES

[1] Storey M A, Treude C, Deursen A V, et al. The impact of social media on software engineering practices and tools[C]// The Workshop on Future of Software Engineering Research, Foser 2010, at the, ACM Sigsoft International Symposium on Foundations of Software Engineering, 2010, Santa Fe, Nm, Usa, November. DBLP, 2010:359-364.

[2] RusselWinder, GrahamRoberts. Java software development [M]. People Post Press, 2008.

[3] Li Junjie. Maven application in enterprise Java software products [J]. Computer Knowledge and Technology, 2011, 07(7):1562-1565.

[4] Dang Haifeng. Through the Java Web development three Musketeers: Eclipse + Tomcat + Ant integration development [M]. Electronic Industry Press, 2008.

[5] Liu Qiao, Li Yang, Duan Hong, etc. A Survey of Knowledge Mapping Construction Techniques [J]. Computer Research and Development, 2016, 53(3):582-600.

[6] Zanjani M, Kagdi H, Bird C. Automatically Recommending Peer Reviewers in Modern Code Review[J]. IEEE Transactions on Software Engineering, 2016, 42(6):530-543.

[7] Yin G, Wang T, Wang H, et al. OSSEAN: Mining Crowd Wisdom in Open Source Communities[C]// IEEE Symposium on Service-Oriented System Engineering. IEEE Computer Society, 2015:367-371.

[8] Yang Bo, Yu Xi, Zhang Wei, etc. Correlation Analysis of Influencing Factors in GitHub Open Source Software Development Process [J]. Journal of Software, 2017, 28(6):1330-1342.

[9] Jiang Ri Nian, Lin Xia, Qiao Dexin. The Introduction and Application of Maven in Java Project[J]. Computer Knowledge and Technology, 2013(21): 4842-4847.

[10] Li Junjie. Application of Maven in Enterprise Java Software Products[J]. Computer Knowledge and Technology, 2011, 07(7): 1562-1565.

[11] Huang Decai, Yan Huachun. PageRank algorithm research [J]. Computer Engineering, 2006, 32(4):145-146.

[12] Wen Tao, Zhu Min, Zhou Ke, etc. Improvement of PageRank Algorithm Based on Multi-weighting Factor [J]. Microcomputer Information, 2012(9):422-424.

[13] Ishii H, Tempo R. Distributed Randomized Algorithms for the PageRank Computation[J]. IEEE Transactions on Automatic Control, 2010, 55(9):1987-2002.

[14] Ren Xiaolong, Lv Linyuan. A Survey of Ranking Methods for Important Nodes in Network [J]. Chinese Science Bulletin, 2014(13):1175-1197.

[15] Guzman E, Li Y. Sentiment analysis of commit comments in GitHub: an empirical study[J]. 2014, 8(399):352-355.

[16] Yang Cheng, Fan Qiang, Wang Tao, etc. Individualized recommendation method for open source projects based on multidimensional features [J]. Journal of Software, 2017, 28(6):1357-1372.

[17] Liu Jinhong, Lu Yuliang. A review of the topic web crawler research [J]. Application Research of Computers, 2007, 24(10):26-29.