Decision Support Systems 91 (2016) 103–112

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Reputation in an open source software community: Antecedents and impacts

Yuanfeng Cai [a], Dan Zhu [b,*]

[a] Zicklin School of Business, Baruch College, City University of New York, New York, NY, United States
[b] College of Business, Iowa State University, Ames, IA, United States

ARTICLE INFO

Article history:
Received 1 July 2015
Received in revised form 11 June 2016
Accepted 3 August 2016
Available online 11 August 2016

Keywords:
Open source software
Positive evaluation
Reputation management
Project success

ABSTRACT

A developer's reputation in the OSS community is determined by all the evaluations received from his or her peers. While a large body of studies focuses on the importance of developers' reputations in their participation motivations, there is still lack of understanding for two issues. First, which factors can lead to a high developer's reputation? Second, how does the overall reputation of the developers' in a project impact project success? In this study, we develop a theoretical model and conduct an empirical analysis in a large online open source community. The results show that a developer's reputation level is determined by his or her 1) coding quality, 2) the deviation of the commitment behavior, 3) community experience, and 4) collaboration experience. In addition, we find that the group with an overall higher level of reputation would achieve a better performance, while the individual reputation level deviation within the group would impair its technical success. The implications of our findings and the future research directions are then discussed.

1. Introduction

Open Source Software (OSS) has been increasingly developed in recent years [90]. It is recognized as a reliable way to produce high quality and cost-effective software [53,63].In contrast to the traditional software development model, the OSS development model heavily relies on the contributions of volunteers, rather than traditional employees [64]. The developers volunteer to devote their efforts and collaborate with each other in online OSS communities, such as Sourceforge and Ohloh [36]. Many projects, such as the Linux operating systems, the Mozilla browser and the Apache web server have been successfully developed in OSS communities [64].

Despite the successful cases, it is also observed that a large number of OSS projects end in failure [42]. One important cause for their failure is a lack of sustained contributions [25,50]. Since OSS developers are self-motivated to contribute in OSS communities, it is unlikely to adopt traditional methods, such as employment contracts and monetary incentives, to keep them motivated to contribute further [50,64]. Herein, understanding the motivation mechanism behind OSS developers is necessary.

Substantial amount of studies have explored developers' motivation for OSS project participation. A variety of motivations have been identified, such as altruism, career, enjoyment and learning [32,33,40,41,43,76,85,91]. Among these quoted motivations, a vital one is a developer's desire for reputation [17,30,33,41,43,50].

Reputation is a distribution of opinions, estimations, or evaluations about an entity, such as people, item and organization, in an interest group [12]. An interest group is a group in which people concern for one another. In the OSS community, developers can benefit by working with an intelligent, experienced and generous developer [63]. Hereby, a developer's reputation, which can attract future collaborations and attentions, is concerned in the community [46]. In the OSS community, developers can evaluate each other [36] and the average of all received peer evaluations is a developer's reputation. A developer with more positive evaluations can attain a higher reputation. The desire for reputation building attracts developers to join in and effectively contribute to the OSS community [70].

Despite many previous studies on the importance of reputation, very few of them center on the antecedents of the reputation and they produce contradictory results. For example, Roberts et al. [64] discovered that a developer's reputation relies on his or her contribution, while Hu et al. [36] documented that homophily factors, regardless of a developer's contribution, are most influential. One possible reason for this inconsistency is a lack of systemic theoretical foundation. In addition, previous studies claim that the evaluation is affected by the existing reputation, which limits their applicability to developers who haven't received any evaluation [36,75]. Given the limitations in the literature, it raises an unanswered and vital question that we will address in this study: Which factors are the antecedents of a developer's reputation in the OSS community, i.e. how to attain a good reputation? In answering this question, we adapt the theoretical framework proposed by Homans [35] and collect a dataset from a large OSS community to delve into the antecedents both theoretically and empirically. The answer to

* Corresponding author.
E-mail addresses: Yuanfeng.Cai@baruch.cuny.edu (Y. Cai), dzhu@iastate.edu (D. Zhu).

http://dx.doi.org/10.1016/j.dss.2016.08.004

this question is imperative, as insufficient participation has long been considered as a challenging issue in the OSS community [4,15]. Understanding the antecedents can help developers build better reputation, which can motivate developers' continuous participation in the OSS community [64].

Reputation motivates developers' participation in the OSS development [64], but it is unclear whether it affects the project performance. Many other factors critical to OSS success have already been uncovered in previous research, such as the project license type [44,79], technological factors, e.g., the programming language and developers' activity [19, 52,79], the project network structure [21,29,73], the diversity of the development team [20], intellectual property rights [90], and prior collaboration experiences [30]. Although several studies suggest that project success is affected by developers' reputation-based motivation [11,36,78,92], there is still a lack of empirical evidence. Thus, our second research question addresses the relation between the level of reputation and project success: For one project, how can its overall level of the developers' reputations affect its performance? The answer to this question has implications for project managers. For example, in the OSS community, developers work collaboratively to develop a project [25]. In many communities, such as Ohloh, a developer's reputation can be easily observed. By understanding the influence of reputation on project performance, project managers can better manage participants and thereby achieve a better performance. Tirole [82] argues that the team reputation can be modeled as an aggregate of individual reputations. For a project with multiple developers, its overall reputation level is termed *group aggregated reputation*. Following the definition in previous literature [38,55], this reputation level is measured as the weighted average of all its members' individual reputation. We relate group aggregated reputation of a project with its performance to empirically examine their relationship.

Our study contributes to the OSS literature mainly in two ways. First, it proposes a comprehensive model to investigate factors impacting a developer's reputation in the OSS community. Different from the prior studies, we do not assume that developers must already have a certain level of reputation. Hereby, findings from this study can be generalized to all developers in the OSS community. Second, this research explores whether developers' reputations affect the success of a project. To the best of our knowledge, this study is the first to empirically examine the role of reputation on project performance. Previous studies have demonstrated the influence of a developer's reputation on his or her motivation [30,43,64], but they seldom connect it with project performance. Considering the relatively large failure rate of OSS projects [42], our findings can provide insight to project managers regarding the improvement of project performance.

We begin by reviewing the related literature in the next section. Section 3 presents the theoretical framework and develops the associated hypotheses. In Section 4, we introduce details about the dataset we will use in the empirical evaluation. Results and findings are discussed in Section 5. We conclude the paper in Section 6 by identifying the contributions, limitations and implications of this study.

2. Related literature

With a growing popularity of OSS communities, extensive research has been carried out in this domain. In this study, we mainly draw on two streams of research pertaining to the OSS community: 1) OSS developers' motivation and reputation, and 2) OSS project performance. Built upon the related studies, we identify the research gap at the end of this section.

2.1. OSS Developers' motivation and reputation

The OSS community cannot thrive without the voluntary contributions from the self-motivated OSS developers [64]. Hence, scholars have long been interested in the exploration of developers' motivations. Developers' motivations are normally categorized into three groups: intrinsic, internalized extrinsic, and extrinsic motivations [86]. Intrinsic motivations are psychological factors related to developers' internal needs for self-satisfaction, such as altruism and enjoyment [32,33,40, 41,64,68,91]. Extrinsic motivations are generally external compensations such as career opportunities ([32,33,41,43,64,91]). Internalized extrinsic motivations are internalized external incentives for developers' self-regulation, such as reputation and learning [17,30,32,33,41, 43,50,64,76,85,91,92]. This research zeroes in on the reputation-based motivation, which is found as a major drive for developers' participation in OSS projects [25,79].

In the cyber world, developers receive evaluations through either content-driven reputation systems or user-driven reputation systems [13,38]. Content-driven reputation systems are applied to wiki-based websites, such as Wikipedia. They derive the reputation score based on contributors' content evolution, that is, contributors gain high reputation if their content can last longer [1]. The other type, user-driven reputation systems, is popularly applied to electronic-commerce websites, such as Amazon. Customers on Amazon can share their feedback in the form of numerical ratings towards entities, e.g., sellers at Amazon. A seller's reputation score is calculated as the average of all ratings [67]. OSS communities, such as Ohloh, adopt user-driven reputation systems, in which peers evaluate developers based on their knowledge contributions [20,46,63,70]. A developer's reputation score is determined by all peer evaluations, as a reflection of an overall peer recognition [25,88].

While the importance of reputation has been acknowledged in previous studies, research on the antecedents of a developer's reputation has been fragmented and ambiguous. Stewart [75] discovered that a developer's current reputation has an impact on his or her future reputation. This finding is not meaningful for developers without any reputation. Roberts et al. [64] demonstrated that developers' contributions, e.g. commits, play an important role in their reputation. However, Hu et al. [36] uncovered that developers' coding contribution do not affect their reputations. Instead, a developer is likely to receive a positive evaluation from a peer if 1) his or her current reputation is high, 2) he or she has mutual acquaintances with the evaluator and 3) he or she shares homophily in location, nationality, programming language, and community status with the evaluator [36]. Consequently, findings from the extant literature are limited and inclusive. A systematic examination on factors affecting a developer's reputation is essential.

2.2. OSS project performance

With the unprecedented prosperity of OSS projects, scholarship has leaned towards exploring factors affecting project success. However, extant studies lack an objective measure of OSS project success [52]. Since its development relies on voluntary participations, the indicators for the traditional software success, such as being on-time, being on budget, and meeting specifications cannot be applied to the OSS project [78]. Some measures such as use and use environment are utilized widely to measure information system project success [22,42]. However, they are not observable in the OSS setting [18]. Therefore, other measures have been designed in the previous studies. For example, OSS project success can be measured as the total amount of contributions from all developers [50,77]. In a similar vein, community size and bug-fixing time are considered as correlated and important measures for OSS project success [18,52,63,73]. Additionally, English and Schweik [23] classified successful projects based on their number of file releases and Stewart et al. [78] utilized the number of the user mailing list subscription as a measure for OSS project success. The aforementioned measures are relevant to both the project development process and the developer's contributions, which are publicly visible in the OSS community [18]. However, Grewal et al. [29] brought up that it is incomplete to measure project success only in terms of its technical achievements. In congruence with the literature on software success [62] and that on new product development [49], they proposed a two-dimensional measure for OSS success, which contains market success, defined as the level of users' interest, and technical success, defined as the number of commits. We adopt this measure in our analysis of project performance in the following sections.

A variety of factors have been found to affect project success in previous studies. For example, Daniel et al. [20] analyzed how developers' diversity affects their project performance. Wen et al. [90] investigated the impact of intellectual property rights enforcement on project performance. Furthermore, it was discovered that the less restrictive license can attract consistent contributions and result in more successful projects [44,52,79]. The developers' prior collaborative experiences are believed to affect project's success [21,29,30,73], and projects with a more common programming language are more likely to be successful [19,79]. In addition, projects with a large number of active developers can keep sustained contributions and accordingly engenders the successful projects [52,54,79]. Despite extensive studies on determinants of success, few of them empirically examine the relationship between developers' reputations and project performance.

An assessment of the literature reveals a universally acknowledged role of each developer's reputation in his or her participation motivation. Notwithstanding this sufficient recognition, however, two research gaps remain. First, with the importance of a developer's reputation, no coherent discourse has emerged regarding the antecedents of a developer's reputation. Second, the literature is silent regarding the effects of developers' reputations on project performance. We aim to fill these voids in this study.

3. Theoretical framework

In this section, we first examine the antecedents of a developer's reputation by following Homan's theoretical lens in social behavior. We then further explore whether reputation has an impact on project performance.

3.1. Antecedents of a Developer's reputation

The theoretical foundation for the antecedents of community reputation is adapted from Homans's [35] studies of small groups. As shown in Fig. 1(a), Homans [35] suggests that all group behavior consists of three interrelated elements: activities, interactions, and sentiments. Activities are the tasks people need to do; Interactions are any types of communications between two or among more people; Sentiments are attitudes from one person to the other [35]. According to Homans [35], a person's attitude towards another person depends on the evaluatee's activity as well as their interaction. The evaluation a person receives will be determined by: 1) whether his or her activities are closer to his or her expected job duties, i.e. job performance; 2) his or her interaction experience with the evaluator. In this study, we adapt this framework to the OSS setting, where developers are responsible for converting project requirements into code. Two types of developers' factors, their coding performance in the OSS community, and their interaction experiences with peers may significantly influence developers' community reputation, as shown in Fig. 1(b). Thus, we hypothesize that these two types of factors are the antecedents of a developer's community reputation.

3.1.1. Coding performance

Homans [35] argues that a person can achieve a higher reputation if his or her activities are closer to the norm. The norm is what a person is expected to complete [35]. In the OSS community, the primary responsibility of developers is building and maintaining the software, i.e., writing code [64]. Hereby, a developer's coding performance is defined as whether he or she can create codes that meet the needs of the OSS projects. A developer's coding performance is expected to positively affect his or her reputation.

We measure a developer's coding performance from two dimensions. First, previous studies suggest that OSS developers are expected to contribute high-quality codes [7,24,43,68,86,87,93]. Thus, coding quality is one critical basis on which evaluators make their evaluations. Coding quality, which indicates the programming skill to complete the tasks, is hard to directly measure in most OSS communities [7]. However, developers' coding quality can be assessed through the user ratings of projects they have already produced [51,74]. Therefore, we use the user ratings of previous projects as a proxy for developers' coding quality [71]. The working credibility of a developer is signaled by his or her prior experiences and prior performance [45]. The more satisfactory work a developer has produced, the more likely that he or she can receive a positive evaluation. Scholarship has pointed out that prior group project reputation can benefit its group member's individual reputation [3,48,82]. Developers that have participated in a number of well-received projects can increase their perceived contribution credibility [6], so that peers can have a relatively high confidence in their coding quality. Meanwhile, developers seldom have face-to-face communication opportunities in the OSS community and thus are not well acquainted with their peers [20]. User ratings of projects are observable in most OSS communities. Thus, it is one of the most convenient sources from which developers can learn about their peers. Evaluators do not have to go through each line of code, but instead refer to ratings easily when making evaluation decisions. While each individual in one group can inherit a priori reputation on the basis of that group's reputation [81], the influence of the previous group reputation on each individual reputation should not be the same. Developers in the same project do not contribute equally. Some developers initiate projects and work on them thoroughly, while others join in last minute. Accordingly, they are not supposed to take the same credit from their group's prestige. A developer's participation level can be quantified by his or her commits out of the entire project's commits. Thus, coding quality is defined as the weighted average of previous projects' ratings, where the weight depends on the individual's contribution ratio. A high level of coding quality indicates a decent past coding performance. A developer's reputation can be improved through consistently quality

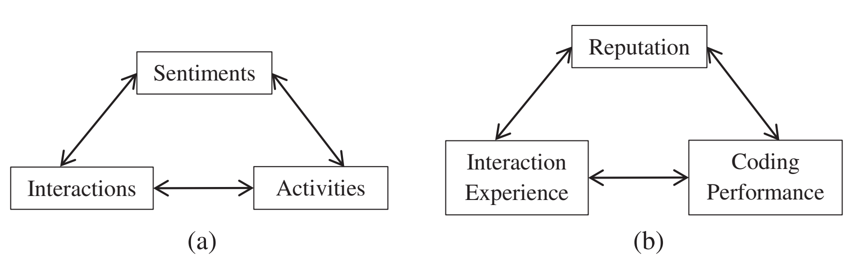

Fig. 1. (a) Homans [35] Theoretical Lens (b) an adaption of Homans to OSS Community.

contributions [70]. An excellent prior performance record means a low risk of norm violations. According to Homans [35], a developer who makes contributions better satisfying users' needs is more likely to be appreciated by peers. Hence, we hypothesize:

H1a. *Coding quality is positively related to a developer's reputation.*

The second dimension for developers' coding performance is their commitment behavior. Commitment behaviors relate to the process in which an individual commits to a certain group and intends to stay in that group [65]. Due to their voluntary participation nature, developers can stay or leave a project at any time [17,25]. They are not restricted by employment contracts [64]. Therefore, different developers can show distinct commitment behaviors. Some developers consistently engage with the OSS community. They have a relatively stable contribution pattern, which makes a similar amount of commits in each period. Other developers contribute to the community in a random manner [26]. As illustrated in Fang and Neufeld [26], such developers can contribute intensively in the beginning, and then become inactive for several quarters before making the next contribution. Consequently, they have a high variability of their quarterly contribution. We capture a developer's commitment behavior by the standard deviation of periodic commitment. A larger deviation indicates a higher degree of randomness in the developer's behavior. The standard deviation of commitment behavior is an important dimension of a developer's coding performance. Shah and Kruglanski [69] find that the individual's reputation is related to the frequency of doing the activity. A developer is expected to provide the high quality, sufficient and on-going support for project development [80]. Previous studies found that developers' unstable commitment behavior is an impediment to effective sharing of knowledge with other peers [37,56]. In addition, the development of OSS projects relies on developers' consistent contributions [15,25]. A large percentage of OSS projects fail due to a lack of participation [33]. As OSS projects desire a continued level of coding activity from voluntary developers [16], a developer with ongoing contributions can better satisfy their needs and behave closer to his or her expected role. According to Homans [35], peers should more positively evaluate such developers. It means that the standard deviation of commitment behavior negatively affects a developer's reputation. Thereby, we propose the following hypothesis:

H1b. *Deviation of commitment behavior is negatively related to a developer's reputation.*

3.1.2. Interaction experience

Homans [35] argues that a person can achieve a higher reputation if he or she has more interaction experiences with peers. While it lacks face-to-face interaction opportunities in the OSS community [66], developers can still accumulate interaction experiences through an active participation in the community [30]. For example, developers can interact with peers through community response and online discussion to ask or solve questions from other members in the community [93]. Previous project collaboration experiences also provide communication opportunities between peers [30]. Homans [35] defines interaction as a long and frequent communication opportunity between peers. In most OSS communities, it is not easy to calculate the accurate time a developer spent due to a lack of log in and log out record. Community experience, which is defined as the total number of months which a developer made a contribution, could be a proxy for his or her potential interaction duration with peers. We do not adopt the developer's "age" in the community, i.e. the total months since the developer has joined in, because a developer is not always active.

Community experience is expected to positively contribute to a developer's reputation. A longer active time in the community can facilitate knowledge sharing with peers, from which developers can benefit themselves [28,61]. In the OSS community, developers can help each other solve problems through online response or discussion forums [10]. The more time a developer obtains a help from other peers, the more knowledge he or she learns from peers; the more time a developer spends replying to others' questions, the more information he or she shares with others [27]. An effective knowledge sharing process requires a smooth interaction, which further positively contributes to reputation attainment. Moreover, a longer active participation can increase a developer's visibility as well as the interaction chance. Following Homans [35] topology, such a developer is more likely to be positively evaluated. We hypothesize:

H2a. *Community experience is positively related to a developer's reputation.*

The other dimension of communication, communication frequency, is measured by collaboration experience. Collaboration experience is the total number of peers a developer has ever collaborated with. A rich collaboration experience provides a developer with opportunities to learn from peers. Developers can gain various types of experience and knowledge by collaborating with different people. They can also access peers' skill sets so that their own knowledge bases can be expanded [47]. A developer with stronger collaboration experience is more likely to interact with other developers effectively, either by contributing his or her knowledge or by solving others' problems. In addition, past collaboration experience can result in greater trust between peers [83]. Developers are more inclined to work with someone they have worked with in the past [30]. Hereby, as suggested by Homans [35], collaboration experience is expected to positively contribute to reputation attainment. The hypothesis is developed as below and the research model for the antecedents of reputation is shown in Fig. 2.

H2b. *Collaboration experience is positively related to a developer's reputation.*

3.2. Reputation and project performance

In this subsection, we explore how developers' reputations can affect their project performance. For a project developed by more than one developer, its development group reputation can be modeled as an aggregate of individual reputations [82]. Consistent with the definition in the literature [38,55], *group aggregated reputation* is defined as the weighted average of all group members' individual reputation. For a development team with more than one developer, whether the difference in their developers' reputations affects project performance is also of our interest. As suggested by Harrison and Klein [31], we term the reputation difference in one team as *reputation disparity diversity*, which measures the diversity of developers' reputations.

3.2.1. Group aggregated reputation

Prior research suggests that reputation has a positive impact on project performance [36]. Since reputation is an important type of participation motivation [64], developers with a high reputation contribute actively to projects. In the OSS community, developers' motivations are essential to project performance [78,92]. When there are pending tasks, developers with higher reputations are more likely to actively carry them out [64]. Thus, pending tasks in a group with multiple high-reputation developers can be undertaken by more than one developer. Contribution redundancy allows choice among a variety of solutions, which increases the quality of the solution and project performance [11]. Meanwhile, a high level of individual reputation is found to drive the developers' collective spirit, which encourages them to persist in working and putting in effort to solve problems [39]. Therefore, a group with a larger group aggregate reputation, which consists of more well-reputed developers, is more likely to create better projects.

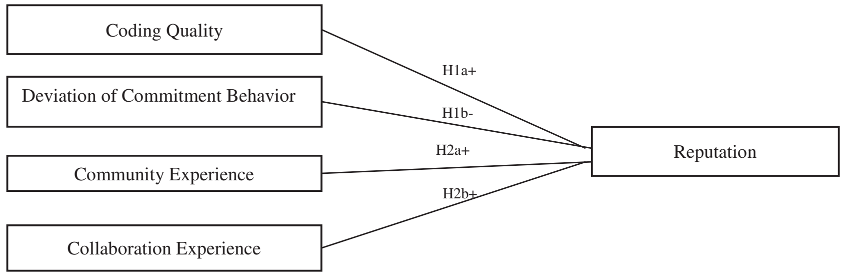

Fig. 2. Research Model for Antecedents of Reputation.

Furthermore, a developer's high individual reputation reveals a reliable contribution behavior, as reflected by peers' evaluations. If a developer leaves a project without completing his or her tasks, his or her reputation should be negatively affected. Hereby, developers with a higher reputation are more likely to accomplish the tasks in a responsible manner. Thus, a high level of group aggregated reputation should positively contribute to the project performance. Hereby, we propose the following hypothesis:

H3. *Group aggregated reputation has a positive impact on the group's project performance.*

3.2.2. Reputation disparity diversity

Harrison and Klein [31] have conceptualized three fundamental types of diversities among team member: disparity, separation, and variety. Disparity diversity captures differences among team members in status, seniority or prestige. Separation diversity refers to differences in opinion among team members, primarily of attitudes, cultures, and beliefs. Variety diversity reflects team members' possession of different information and experiences. Following their topology, in one project development team, the difference in reputation among team members, is the reputation disparity diversity. In this subsection, we discuss its impacts on project performance. The impact of the other two types of diversities on project performance is discussed in the later section as control variables.

A group with a large value of reputation disparity diversity indicates its developers' reputations inequality. Some developers are well-reputed, while others are not. Previous studies uncovered that developers are aware of the reputation diversity in their development team [84]. Team members with a higher status may limit the participation from lower status members, and therefore negatively affect the quality of interpersonal relationships within the group [60]. Team performance can be maximized only through collaboration, while reputation disparity may hurt it [14]. Developers with better reputations can receive more contribution opportunities [89]. Meanwhile, efforts from low-reputation developers are less visible and those new contributions are less likely to be taken into consideration ([27]). Project performance will decline if it cannot combine the efforts from all group members and take advantage of the necessary inputs from low-status members [14]. It is widely recognized that reputation disparity incites deviances among team members and it can result in silence and suppression of creativity [9,72]. Accordingly, a large value of reputation disparity diversity is expected to deteriorate project performance. In contrast, a group with less reputation disparity is relatively more cohesive, since they are more similar and attractive to each other [59]. Group cohesion is important for group performance, especially in the virtual community, which lacks face-to-face communication opportunities [5]. Herein, a hypothesis is developed as below and the research model for the influence of reputation on project performance is shown in Fig. 3.

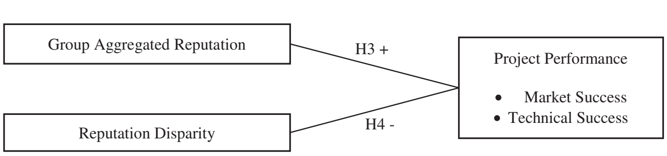

Fig. 3. Research Model for the Influence of Reputation on Project Performance.

H4. *Reputation disparity diversity has a negative impact on group's project performance.*

4. Dataset

In this section, we introduce the dataset used for the empirical testing and the operationalization of dependent and independent variables.

4.1. Data source

We collect a dataset from a large online OSS community—Ohloh [57]. Ohloh retrieves various types of information about OSS projects and developers. For projects, it collects their statistics such as developers, users, user ratings, monthly commits, monthly lines of codes, and programming languages. The user rating for each project is a numerical value from 1 to 5, with a larger value indicating a more satisfactory project. For developers, it tracks their personal information, monthly participation, programming language and evaluations. Ohloh analyzes publicly available information about OSS projects and developers, but it does not host projects [36].

One advantage of Ohloh is that it provides developers' peer evaluation information, which fits our research requirements. Each developer can send a "Kudo" to show his or her appreciation or respect to another developer. In the Ohloh community, a developer can send the "Kudo" link to any other peer no matter if they ever participate in the same project. A "Kudo" expresses a positive evaluation from the sender, in recognition for the receiver's performance and contribution. Given the quantity of all received "Kudos", a reputation score is calculated and updated periodically for each developer. The reputation score is an integer ranging from 1 to 10. A high reputation score indicates that this developer is widely recognized by peers within the Ohloh community.

We develop a set of programs to automatically query and retrieve information [58] for 262 projects with at least two developers each. For each project, all its developers' identification information is recorded. Then we retrieve the developers' information for all developers participated in those 262 projects. For each developer, we collect the reputation score as well as the information related to their development activity, e.g. commits, programming language preference and project participation. The final dataset contains the development-related information for 2496 developers.

4.2. Operationalization of variables

4.2.1. Dependent variable

A developer' community reputation is measured directly from his or her own reputation score. Following previous researchers [29], we measure OSS project performance in a more comprehensive way by constructing a pair of variables. Grewal et al. [29] suggest that OSS project performance can be measured from two aspects: market success, defined as the level of users' interest, and technical success, defined as the number of commits. We operationalize market success as the number of users and technical success directly as the total number of commits. In the OSS projects, some of its users can be its developers at the same time. To be more objective, we exclude those who are its developers when counting the number of users for each project.

4.2.2. Independent variable

As discussed in Subsection 3.1, for each developer, we construct four independent variables using Ohloh data related to his or her coding performance and interaction experiences.

- Coding quality: the weighted average of the rating values from all a developer's participated projects. The weight depends on the developer's contribution percentage in each project. In one project, the contribution percentage is calculated as an individual's commits out of the total commits.
- Deviation of commitment behavior: the standard deviation of monthly commits.
- Community Experience: the number of months that a developer has made at least one commit in the community.
- Collaboration experience: the number of different peers who the developer has collaborated with in the same project.

For each project, we construct two independent variables related to its developers' reputations.

- Group aggregated reputation: calculated as the weighted average of all its developers' individual community reputation scores. We weight a developer's reputation by the number of months he or she has spent on the project and assign a higher value to developers with a longer contribution.
- Reputation disparity diversity: calculated as the standard deviation of all developers' individual community reputations.

4.2.3. Control variable

In the analysis of H1 and H2, we control the number of projects a developer has participated in. In the analysis of H3 and H4, we control the following variables:

- Previous Collaborations: For each group, we calculate the group member collaboration experience as $\frac{(\text{Number of Previous Collaboration})}{N(N-1)/2}$. The numerator is the number of times that two developers have ever participated in the same project. The denominator is the number of all possible combinations between two developers, where N is the developer size within a group.
- Software Size: As defined and controlled in Colazo and Fang [15], it is measured as the total lines of code of each project.
- Project Age: It is calculated as the amount of time (in number of months) since a project was registered in the community.
- Developer size: It is the number of developers participated in this project.

Table 1(a)
Variable descriptive statistics for the model shown in Fig. 2 (N = 2496).

Variable	Mean	Std. dev.	Min	Max
Reputation	8.284	1.042	1.000	10.000
Coding quality	0.924	1.496	0.000	7.127
Deviation of commitment behavior	6.875	4.632	0.000	18.043
Community experience	52.236	42.328	1.000	167
Collaboration experience	505.33	668.795	10	4500
Number of projects	18.337	37.236	1.000	380.000

Table 1(b)
Inter-correlations for the model shown in Fig. 2 (N = 2496).

	Variable	1	2	3	4	5	6
1	Reputation	1.000					
2	Coding quality	**0.27**	1.000				
3	Deviation of commitment behavior	**-0.18**	0.01	1.000			
4	Community experience	**0.51**	0.23	-0.01	1.000		
5	Collaboration Experience	**0.17**	0.0	0.02	**0.23**	1.000	
6	Number of projects	**0.27**	0.21	-0.01	**0.39**	**0.35**	1.000

*bold indicates p < 0.05.

- Number of Programing Languages: Each developer has his or her own primary programming language. The number of programming languages is defined as the total number of distinct primary languages across all group members.
- Culture separation diversity: As discussed in Subsection 3.2.2, we control the other two types of diversities among team members. One of them is separation diversity. Previous studies suggest that a major type of separation diversity in the OSS community is cultural separation, which can be measured as differences in home country among team members [2, 34]. As country is a categorical variable, we use Blau's index [8] to calculate the diversity. Blau's index is calculated as $1-\sum p_k^2$, where p is the proportion of members in the K^{th} category.
- Tenure variety diversity: The last type of diversity is tenure variety, which captures differences in tenure among team members. Tenure is the number of months since a developer registered in the community. For one project, tenure variety is the standard deviation across all developers' community tenure.

5. Empirical analysis and results

This section reports on findings from the empirical analysis. Using a stepwise multivariate regression model of analysis, we arrived at a set of results to interpret the relationship between dependent variables and independent variables. Tables 1(a), 1(b) presents the descriptive statistics and summarizes variable correlations in the examination of the antecedents of an individual's reputation, i.e., the model shown in Fig. 2. The Durbin Watson value for this model is below 3, i.e., 1.65. Tables 2(a), 2(b) displays the descriptive statistics and variable correlations in the examination of the influence of reputation on project performance, i.e., the model shown in Fig. 3. Since project performance is captured by two variables: the number of users and the number of commits, two models are examined for project performance. One is for market success and the other is for technical success. The Durbin Watson values for both models (i.e. market success and technical success) are below 3 (i.e., 1.98 and 2.00 respectively). Results from Tables 1(b) and 2(b) suggest that multicollinearity is less likely to be a

Table 2(a)
Variable descriptive statistics for the model shown in Fig. 3 (N = 262).

Variable	Mean	Std. dev.	Min	Max
Market Success	191.95	488.40	0	4482
Technical Success	12,289.98	16,793.21	246	94,494
Group Aggregated Reputation	8.47	0.79	1.45	9.92
Reputation Disparity	0.77	0.63	0.00	5.65
Software Size	830,061.03	2,358,064.05	2433	23,070,442
Project Age	125.28	27.83	24	165
Previous Collaborations	0.46	0.55	0.00	4.13
Developer Size	9.38	14.95	2	116
Number of Language	1.74	1.00	0.00	8.00
Tenure Variety	8.88	16.70	0.00	22.02
Culture Separation	0.53	0.29	0.00	1.00

Table 2(b)
Inter-correlations for the model shown in Fig. 3 (N = 262).

	Variable	1	2	3	4	5	6	7	8	9	10	11
1	Market Success	1										
2	Technical Success	**0.27**	1									
3	Group Aggregated Reputation	**0.22**	**0.06**	1								
4	Reputation Disparity	**-0.19**	**-0.08**	0.05	1							
5	Software Size	**0.32**	**0.4**	0	-0.08	1						
6	Project Age	-0.12	0.08	-0.04	0.01	**-0.15**	1					
7	Previous Collaborations	**-0.17**	**-0.3**	-0.08	-0.09	**-0.15**	-0.01	1				
8	Developer Size	**0.24**	**0.54**	0.11	-0.02	**0.1**	0.05	**-0.26**	1			
9	Number of Language	**0.23**	**0.24**	0.11	-0.02	0.17	0.04	-0.23	0.51	1		
10	Tenure Variety	0.01	0.09	0.12	**0.17**	0.03	0.2	-0.24	-0.02	-0.01	1	
11	Culture Separation	**0.20**	**0.2**	**0.12**	-0.02	0.1	-0.01	-0.24	0.38	0.3	-0.08	1

*bold indicates $p < 0.05$.

concern for all three models. The Durbin Watson tests results show that there is no evidence of serial correlation.

Table 3 shows the results for the model shown in Fig. 2. The coefficient for the coding quality (0.088, $p < 0.001$) indicates that users' ratings of a developer's prior work has a positive impact on his or her reputation. If a developer has participated in a number of well-received projects, he or she can inherit a positive influence from his or her previous work. Accordingly, he or she is more likely to be positively recognized by peers. Thus, H1a is supported. The negative coefficient of deviation of commitment behavior (−0.001, $p < 0.05$) indicates that it has a negative influence on a developer's reputation. A higher level of the variability of a developer's monthly contributions can restrain him or her from receiving positive evaluations. Thus, H1b is supported.

Results in Table 3 also show that two factors related to a developer's interaction experience, community experience (0.012, $p < 0.001$) and collaboration experience (0.001, $p < 0.001$), have significant positive impacts on a developer's reputation. A developer with a longer working length is found to attain a higher reputation in the OSS community. Developers who have collaborated with more peers are more likely to receive positive evaluations. Therefore, H2 is supported. As expected, the number of projects a developer has ever participated in has a positive effect on his or her reputation. Although their coefficient values are not large, community experience and collaboration experience still significantly affect a developer's reputation score. Because the values of community experience and collaboration experience are well above that of developers' reputation, they significantly affect a developer's reputation value even with the relative small coefficients.

Findings from Table 3 have demonstrated the applicability of the Homans [35]'s theoretical framework to the OSS community. Interaction experiences and community performance facilitate a community member's reputation. Our findings indicate that developers with a longer active participation period have advantages over the newly joined members in reputation building. However, both of them can boost their reputation by participating in more projects, especially those with a larger developer size. At the same time, they should contribute continuously and intensively, so that they can inherit more credits from their prior work. The results also solve the inconsistent findings of the antecedents of the developer's reputation in previous studies [36,64] and provide both theoretical and empirical support.

Table 4 displays the results from the model shown in Fig. 3. Results show that the overall reputation level of a development team has a significant positive impact on both market success and technique success. Thus, a project from a development team with a higher group aggregated reputation is more likely to be widely adopted. Meanwhile, a project with a group of highly motivated developers can attract more contributions. Thereby, H3 is supported. In addition, a development team's reputation disparity has a significant negative relationship with its project's technical success. A team with a more homogeneous level of motivation is more likely to attract developers' contributions. However, the relationship between reputation disparity and project's market success is insignificant. Thus, a project's reputation disparity does not significantly affect its number of users. Hence, H4 is partially supported. This finding may be partly attributed to that users are relatively difficult to observe the group's reputation disparity directly. However, they can more easily access the overall reputation level as each developer's reputation is available in the community. In addition, the number of developers in one team affects its project performance. The number of primary programming languages and the time length of the project also contribute to the technical success of a project.

Findings from Table 4 have empirically validated the conjectures in previous studies [11,36,78,92], that is, project performance is related to the reputations of project developers. Our results have shown that a project developed by a number of highly-reputed developers is much likely to be adopted by users. This finding implies that a project manager should invite developers with higher reputations if the manager intends to increase the market acceptance rate of the project. Furthermore, developers in a group with a number of highly reputed developers are more motivated to contribute. Thus, results in both Tables 3 and 4 show that a developer's past contributions affect his or her reputation, while a higher reputation can further motivate a developer to continue

Table 3
Results for the model shown in Fig. 2.
(N = 2496, '***'$p < 0.001$, '**' $p < 0.05$, '*'$p < 0.1$)

Variable	Coefficient
Coding quality	0.088***
Deviation of commitment behavior	-0.001**
Community Experience	0.012***
Collaboration Experience	0.001**
Number of Projects	0.002**
R^2	0.328***

Table 4
Results for the model shown in Fig. 3.
(N = 262, '***'$p < 0.001$, '**' $p < 0.05$, '*'$p < 0.1$).

Variable	Market Success Coefficient (Number of Users)	Technical Success Coefficient (Number of Commits)
Group Aggregated Reputation	89.91**	1358.00**
Reputation Disparity	-51.52	-309.20*
Software Size	0.00***	0.00***
Project Age	-1.31	60.30*
Previous Collaborations	-44.46	-566.60
Developer Size	4.05*	593.50***
Number of Language	34.05	-1618.00*
Tenure Variety	0.00	68.16
Culture Separation	128.60	3150
R^2	0.212***	0.400***

Table 5
Results of Hypothesis Testing.

Hypotheses:	Result
H1a (+): Coding quality is positively related to a developer's reputation.	Supported
H1b (−): Deviation of commitment behavior is negatively related to a developer's reputation.	Supported
H2a (+): Community experience is positively related to a developer's reputation.	Supported
H2b (+): Collaboration Experience is positively related to a developer's reputation.	Supported
H3 (+): Group aggregated reputation has a positive impact on project performance.	Supported
H4 (−): Reputation disparity diversity has a negative impact on project performance.	Partially Supported

working. Accordingly, a group with more high-reputation developers can produce a larger quantity of commits. This finding is also consistent with Homans [35]'s argument that activities, interactions, and sentiments are three inter-correlated elements, as illustrated in Fig. 1. Table 5 summarizes the results of hypothesis testing.

6. Conclusions and discussions

Individual reputation has been widely recognized as a critical motivation for developers' voluntary contributions in the OSS community. However, motivation does not necessarily lead to the success of a project, especially when there are multiple developers with diverse levels of motivations in one development team. Previous research is silent about how the level of developers' reputation affects project performance. Furthermore, despite of the importance of reputation, it is unclear about how to attain a high level of reputation. In this study, we use a dataset from a large online OSS community—Ohloh to empirically examine the antecedences of reputation and how developers' reputations impact their project performance.

In the examination of the antecedents of reputation, we adopt the theoretical framework from Homans [35]. Homans [35] suggests that the reputation is affected by the evaluatee's performance in the community as well as his or her interaction with the evaluator. Our empirical analysis has confirmed his arguments. Four determinants of a developer's reputation have been identified. Two of them are related to the developer's working performance: coding quality and deviation of commitment behavior. The other two are related to the developer's interaction experiences: community experience and collaboration experience.

To explore how reputation affects project performance, we define OSS project performance using a pair of measures: technical success and market success. The overall reputation level in one group is defined as group aggregated reputation. A higher level of group aggregated reputation indicates that developers in this group are with a stronger motivation. As expected, it is found to positively affect the technical success of a project. Moreover, the result shows that the diversity of developers' reputation has a significant negative relation with the technical success. One possible reason is that high-reputation developers may limit the contributions from low-reputation ones and thus hurt the collaboration [60]. Therefore, while the developer's reputation can facilitate the individual's contribution, it does not always benefit the project's performance. Furthermore, our results show that a larger group aggregated reputation can facilitate attracting users, while the market success is not affected by reputation disparity. We suggest that reputation disparity does not significantly affect the user size as this information is not observable for users.

Our study has made two theoretical contributions to the OSS literature. First, it has validated that both social interactions and individual contributions are the determinants for a developer's reputation in a both theoretical and empirical way. Previous studies in this direction have left ambiguous findings. For example, studies from social network analysis emphasize the importance of interaction experiences while ignore the effect of working quality [36]. We are one of the first applying Homans [35]'s theory to the analysis in the OSS projects. Our results have demonstrated the adaptability of literature in the traditional group to the online and virtual community. In addition, our results have a broader applicability. They do not have restraints in the empirical analysis that developers must already receive the evaluations. Second, to the best of our knowledge, we are one of the first to empirically evaluate the impact of reputation on project performance. The current research adds to the literature that examines factors affecting project performance. Our findings indicate that reputation, as a motivation, is important to, but not always benefits project performance. This study also provides a set of practical insight. First, as discussed earlier, only a small number of projects can be successful [42]. Since a developer's reputation score is relatively easily observed in the OSS community, project managers can better utilize this information by inviting a homogeneous group of well-reputed developers to the project development. Second, our results delineate a pathway to improve the reputation score. Since a developer's reputation can motivate his or her participation, the identified determinants are important to developers as well as the project manager. The project manager may encourage his or her members by improving their reputations, which can increase the successful rate of the project. Developers can increase their reputation by providing sustained and stable contributions as well as increasing collaboration experiences with peers.

We end our discussions by identifying limitations and future research directions. We select Ohloh as our testbed since it provides the information related to a developer's reputation, which is the focus of our study. Ohloh provides an analysis of the publicly available information about OSS projects. However, as mentioned in Subsection 4.1, it does not host projects. Therefore, several project related information, such as its market structure and modules, is not available in this dataset and is not examined in this study. In our future research plan, we intend to validate the robustness of our models using datasets from other OSS communities. In addition, as noted in Fig. 1, reputation, activities and interaction are inter-correlated. Some studies have analyzed whether a developer's reputation affect their activities [64]. In our study, the results also show that a group with more high-reputation developers will receive more contributions, which is consistent with this relationship. In future studies, we plan to empirically test these two relations: the influence of reputation on developers' job performance and that on interaction experiences. A more comprehensive framework can help us better understand the role of reputation in the OSS community.

Acknowledgements

This research is partially supported by an internal grant from Iowa State University. We are immensely grateful to Professors L. Zhao, A. Schwab and J. George for their guidance and critiques on an earlier version of the manuscript. We thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

References

[1] B. Adler, L. de Alfaro, A Content-Driven Reputation System for the Wikipedia, Proceedings of the 16th International Conference on World Wide Web (WWW), ACM Press, New York, NY, USA 2007, pp. 261–270.
[2] J. Ågerfalk, B. Fitzgerald, Outsourcing to an unknown workforce: exploring open sourcing as a global sourcing strategy, MIS Quarterly 32 (2) (2008) 385–409.
[3] E. Bagheri, A. Ghorbani, Behavior Analysis through Reputation Propagation in a Multi-Context Environment, International Conference on Privacy, Security and Trust, 2006.
[4] R.P. Bagozzi, U.M. Dholakia, Open source software user communities: a study of participation in Linux user groups, Management Science 52 (7) (2006) 1099–1115.
[5] D.J. Beal, R.R. Cohen, M.J. Burke, C.L. McLendon, Cohesion and performance in groups: a meta-analytic clarification of construct relations, Journal of Applied Psychology 88 (6) (2003) 989–1004.
[6] A. Bhattacherjee, C. Sanford, Influence processes for information technology acceptance: an elaboration likelihood model, MIS Quarterly 30 (2006) 805–826.

[7] A. Bianchi, S. Kang, D. Stewart, The organizational selection of status characteristics: status evaluations in an open source community, Organization Science 23 (2) (2012) 341–354.
[8] P.M. Blau, Inequality and Heterogeneity, Free Press, New York, 1977.
[9] M. Bloom, The performance effects of pay dispersion on individuals and organizations, Academy of Management Journal 42 (1999) 25–40.
[10] L. Sproull, S. Kiesler, R. Kraut, Community effort in online groups: who does the work and why, in: S. Weisband, L. Atwater (Eds.), Leadership at a Distance, Lawrence Erlbaum Publishers, Mahwah, NJ 2007, pp. 171–194.
[11] A. Bonaccorsi, C. Rossi, Why open source software can succeed? Research Policy 32 (7) (2003) 1243–1258.
[12] D.B. Bromley, Relationships between personal and corporate reputation, European Journal of Marketing 35 (3) (2001) 316–334.
[13] K. Chatterjee, L. de Alfaro, I. Pye, Robust Content-Driven Reputation, Proceedings of the 1st ACM Workshop on Workshop on Artificial Intelligence Science, New York, NY, USA 2008, pp. 33–42.
[14] A.M. Christie, J. Barling, Beyond status: relating status inequality to performance and health in teams, Journal of Applied Psychology 95 (2010) 920–934.
[15] J. Colazo, Y. Fang, The impact of license choice on open source software development activities, Journal of the American Society for Information Science and Technology 60 (5) (2009) 997–1011.
[16] J. Colazo, Y. Fang, Following the sun: temporal dispersion and performance in open source software project teams, Journal of the Association for Information Systems 11 (11) (2010) 684–707.
[17] K. Crowston, H. Annabi, J. Howison, Defining Open Source Software Project Success, *Proceedings of the 24th International Conference on Information Systems*, Seattle, WA, 2003.
[18] K. Crowston, J. Howison, H. Annabi, Information systems success in free and open source software development: theory and measures, Software Process: Improvement and Practice 11 (2) (2006) 123–148.
[19] K. Crowston, B. Scozzi, Open source software projects as virtual organizations: competency rallying for software development, IEEE Proceedings Software 149 (1) (2002) 3–17.
[20] S.L. Daniel, R. Agarwal, K.J. Stewart, The diverse effects of diversity in global, distributed collectives: a study of open source project Success, Information Systems Research 24 (2) (2013) 312–333.
[21] S.L. Daniel, K.J. Stewart, Open source project success: resource access, flow, and integration, Journal of Strategic Information Systems (2016) (in press).
[22] W.H. DeLone, E.R. McLean, Information systems success: the quest or the dependent variable, Information Systems Research 3 (1) (1992) 60–95.
[23] R. English, C.M. Schweik, Identifying success and tragedy of free/libre and open source (FLOSS) commons: a preliminary classification of sourceforge. Net projects, Proceedings of the First International Workshop on Emerging Trends in FLOSS Research and Development 2007.
[24] J.A. Espinosa, S.A. Slaughter, R.E. Kraut, J.D. Herbsleb, Team knowledge and coordination in geographically distributed software development, Journal of Management Information Systems 24 (1) (2007) 135–169.
[25] Y. Fang, D. Neufeld, Understanding sustained participation in open source software projects, Journal of Management Information Systems 25 (4) (2009) 9–50.
[26] Y. Fang, D. Neufeld, Should I Stay or should I Go? Worker Commitment to Virtual Organizations, In the Proceedings of the 39th Hawaii International Conference on System Sciences, 2006.
[27] S. Faraj, S. Jarvenpaa, A. Majchrzak, Knowledge collaboration in online communities, Organization Science 22 (5) (2011) 1224–1239.
[28] L. Fleming, D.M. Waguespack, Brokerage, boundary spanning, and leadership in open innovation communities, Organization Science 18 (2) (2007) 165–180.
[29] R. Grewal, G.L. Lilien, G. Mallapragada, Location, location, location: how network embeddedness affects project success in open source systems, Management Science 52 (7) (2006) 1043–1056.
[30] J. Hahn, J. Moon, C. Zhang, Emergence of new project teams from open source software developer networks: impact of prior collaboration ties, Information Systems Research 19 (3) (2008) 369–391.
[31] D.A. Harrison, K.J. Klein, What's the difference? Diversity constructs and separation, variety, or disparity in organizations, Academy of Management Review 32 (4) (2007) 1199–1228.
[32] A. Hars, S. Ou, Working for free - motivations for participating in open-source projects, International Journal of Electronic Commerce 6 (2002) 25–39.
[33] G. Hertel, S. Niedner, S. Herrmann, Motivation of software developers in open source projects: an internet-based survey of contributors to the Linux kernel, Research Policy 32 (7) (2003) 1159–1177.
[34] P. Hinds, M. Mortensen, Understanding conflict in geographically distributed teams: the moderating effects of shared identity, shared context, and spontaneous communication, Organization Science 16 (3) (2005) 290–307.
[35] G.C. Homans, The Human Group, Harcourt, Brace, New York, 1950.
[36] D. Hu, J.L. Zhao, J. Chen, Reputation Management in an Open Source Developer Social Network: an empirical study on determinants of positive evaluations, Decision Support Systems 53 (3) (2012) 526–533.
[37] R. Ibrahim, M.E. Nissen, Developing a Knowledge-Based Organizational Performance Model for Discontinuous Participatory Enterprises, Proceedings of the 38th Hawaii International Conference on System Sciences, 2005.
[38] A. Jøsang, R. Ismail, C. Boyd, A survey of rust and reputation systems for online service provision, Decision Support Systems 43 (2007) 618–644.
[39] G.C. Kane, A. Majchrzak, J. Johnson, G. Chen, A Longitudinal Model of Perspective Making and Perspective Taking within Fluid Online Collectives, *Proceeding of International. Conference on Information Systems.* Phoenix, AZ 2009, p. 10.
[40] K.R. Lakhani, E. Von Hippel, How open source software works: "free" user-to-user assistance, Reseach Policy 32 (6) (2003) 923–943.
[41] K.R. Lakhani, B. Wolf, Why Hackers Do What they Do: Understanding Motivation and Effort in Free/Open Source Software Projects, in: J. Feller, B. Fitzgerals, S.A. Hissam, K.R. Lakhani (Eds.), Perspectives on Free and Open Source Software, MIT Press, Boston 2005, pp. 3–23.
[42] S. Lee, H. Kim, S. Guptab, Measuring open source software success, Omega 37 (2009) 429–438.
[43] J. Lerner, J. Tirole, Some simple economics of open source, Journal of Industrial Economics 52 (2002) 197–234.
[44] J. Lerner, J. Tirole, The scope of open source licensing, The Journal of Law, Economics, & Organization 21 (1) (2005) 20–53.
[45] M. Li, A. Kankanhalli, S. Kim, Which ideas are more likely to be implemented in online user innovation communities? An empirical analysis, Decision Support Systems 84 (2016) 28–40.
[46] J. Ljungberg, Open source movements as a model for organising, European Journal of Information Systems 9 (4) (2000) 208–216.
[47] G. Kuk, Strategic interaction and knowledge sharing in KDE developer mailing list, Management Science 52 (7) (2006) 1031–1042.
[48] S. Madnick, N. Prat, Measuring Data Believability: a Provenance Approach, Proceedings of the 41st Annual Hawaii International Conference on System Sciences, 2008.
[49] E. Mansfield, S. Wagner, Organizational and strategic factors associated with probabilities of success in industrial R&D, Journal of Business 48 (2) (1975) 179–198.
[50] M.L. Markus, B. Manville, C.E. Agres, What makes a virtual organization work? MIT Sloan Management Review 42 (1) (2000) 13–26.
[51] R. Me'ndez-Duro'n, Does the allocation and quality of intellectual assets affect the reputation of open source software projects? Information & Management 50 (2013) 357–368.
[52] V. Midha, P. Palvia, Factors affecting the success of open source software, Journal of Systems and Software 85 (2012) 895–905.
[53] A. Mockus, R.T. Fielding, J. Herbsleb, A Case Study of Open Source Software Development: the Apache Server, Proceedings of the 22nd International Conference on Software Engineering 2000, pp. 263–279.
[54] A. Mockus, R.T. Fielding, J. Herbsleb, Two case studies of open source software development: apache and Mozilla, ACM Transactions on Software Engineering and Methodology 11 (3) (2002) 309–346.
[55] L. Mui, Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social NetworksPh. D. thesis Massachusetts Institute of Technology, 2002 (December).
[56] W. Oh, S. Jeon, Membership herding and network stability in the open source community: the Ising perspective, Management Science 53 (7) (2007).
[57] Ohloh, See http://www.ohloh.net
[58] Ohloh API, See http://www.ohloh.net/api
[59] C.A. O'Reilly III, D.F. Caldwell, W.P. Barnett, Work group demography, social integration, and turnover, Administrative Science Quarterly 34 (1989) 21–37.
[60] K.W. Phillips, N.P. Rothbard, T.L. Dumas, To disclose or not to disclose? Status distance and self-disclosure in diverse environments, Academy Management Review 34 (4) (2009) 710–732.
[61] S. Rafaeli, Y. Ariel, Online Motivational Factors: Incentives for Participation and Contribution in Wikipedia, in: A. Barak (Ed.), Psychological Aspects of Cyberspace: Theory, Research, Applications, Cambridge University Press, Cambridge, UK 2008, pp. 243–267.
[62] A. Rai, S. Lang, R. Welker, Assessing the validity of IS success models: an empirical test and theoretical analysis, Information Systems Research 13 (1) (2002) 50–69.
[63] E.S. Raymond, Homesteading the Nooshere, http://tuxedo.org/~esr/writings/homesteading/ 2001.
[64] J. Roberts, I.-H. Hann, S. Slaughter, Understanding the motivations, participation, and performance of open source software developers: a longitudinal study of the apache projects, Management Science 52 (7) (2006) 984–999.
[65] R.D. Iverson, P. Roy, A causal model of behavioral commitment: evidence from a study of Australian blue-collar employees, Journal of Management 20 (1994) 15–41.
[66] W. Scacchi, Understanding the requirements for developing open source software systems, IEEE Proceeding of Software 149 (1) (2002) 24–39.
[67] J. Schneider, G. Kortuem, J. Jager, S. Fickas, Z. Segall, Disseminating Trust Information in Wearable Communities, *Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing*, Bristol, UK, 2000.
[68] S.K. Shah, Motivation, governance, and the viability of hybrid forms in open source software development, Management Science 52 (7) (2006) 1000–1014.
[69] J. Shah, A. Kruglanski, The Structure and Substance of Intrinsic Motivation, in: C. Sansone, J. Harackiewicz (Eds.), Intrinsic and Extrinsic Motivation: the Search for Optimal Motivation and Performance, Academic Press, San Diego, CA 2000, pp. 106–123.
[70] S. Sharma, V. Sugumaran, B. Rajagopalan, A framework for creating hybrid-OSS communities, Information Systems Journal 12 (1) (2002) 7–26.
[71] C. Shen, P. Monge, Who connects to whom? A social network analysis of an online open source software community, First Monday 16 (6) (2011).
[72] P.A. Siegel, D.C. Hambrick, Pay disparities within top management groups: evidence of harmful effects on performance of high-technology firms, Organization Science 16 (2005) 259–274.
[73] P. Singh, Y. Tan, V. Mookerjee, Network effects: the influence of structural social capital on open source project success, MIS Quarterly 35 (4) (2011) 813–829.
[74] I. Stamelos, L. Angelis, A. Oikonomou, G.L. Bleris, Code quality analysis in open source software development, Information Systems Journal 12 (1) (2002) 43–60.
[75] D. Stewart, Social status in an open source software community, American Sociological Review 70 (2005) 823–842.

[76] K.J. Stewart, S. Gosain, The moderating role of development stage in free/open source software project performance, Software Process: Improvement and Practice 11 (2) (2006) 177–191.
[77] K.J. Stewart, S. Gosain, The impact of ideology on effectiveness in open source software development teams, MIS Quarterly 30 (2) (2006) 291–314.
[78] K.J. Stewart, A.P. Ammeter, L.M. Maruping, Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects, Information Systems Research 17 (2) (2006) 126–145.
[79] C. Subramaniam, R. Sen, M. Nelson, Determinants of open source software project success: a longitudinal study, Decision Support Systems 46 (2) (2009) 576–585.
[80] Y. Sun, Y. Fang, K. Lim, Understanding sustained participation in transactional virtual communities, Decision Support Systems 53 (2012) 12–22.
[81] S. Tadelis, Firm reputation with hidden information, Economic Theory 21 (2) (2003) 635–651.
[82] J. Tirole, A theory of collective reputation (with applications to the persistence of corruption and to firm quality, The Review of Economic Studies 63 (1) (1996) 1–22.
[83] B. Uzzi, J. Spiro, Collaboration and creativity: the small world problem, American Journal Sociology 111 (2) (2005) 447–504.
[84] B. Vasilescu, V. Filkov, A. Serebrenik, Perceptions of Diversity on GitHub: a User Survey, 2015 (In *CHASE*).
[85] E. von Hippel, G. Von Krogh, Open source software and the “private-collective” innovation model: issues for organization science, Organization Science 14 (2) (2003) 209–223.
[86] G. von Krogh, S. Haefliger, S. Spaeth, M.W. Wallin, Carrots and rainbows: motivation and social practice in open source software development, MIS Quarterly 36 (2) (2012) 649–676.
[87] G. von Krogh, E. von Hippel, The promise of research on open source software, Organization Science 52 (7) (2006) 975–983.
[88] G. von Krogh, S. Spaeth, K. Lakhani, Community, joining and specialization in open source software innovation, Research Policy 32 (7) (2003) 1217–1241.
[89] S.P. Weisband, S.K. Schneider, T. Connolly, Computer mediated communication and social information: status salience and status differences, Academy of Management Journal 38 (1995) 124–1151.
[90] W. Wen, C. Forman, S. Graham., Research note—the impact of intellectual property rights enforcement on open source software project success, Information Systems Research 24 (4) (2013) 1131–1146.
[91] C.G. Wu, J.H. Gerlach, C.E. Young, An empirical analysis of open source software developers' motivations and continuance intentions, Information & Management 44 (3) (2007) 253–262.
[92] Y. Ye, K. Kishida, Toward an Understanding of the Motivation of Open Source Software Developers, Proceedings of 2003 International Conference on Software Engineering (ICSE2003) 2003, pp. 419–429.
[93] C. Zhang, J. Hahn, P. De, Research note—continued participation in online innovation communities: does community response matter equally for everyone? Information Systems Research 24 (4) (2013) 1112–1130.

Yuanfeng Cai is an Assistant Professor of Computer Information Systems at Zicklin School of Business, Baruch College, City University of New York. Dr. Cai earned her Ph.D degree in Information Systems from Iowa State University. Her major research interests include business analytics, decision-making with imperfect data and open source communities.

Dan Zhu is a professor at the Iowa State University. She obtained her Ph.D. degree from Carnegie Mellon University. Her current research focuses on business analytics and decision support systems. Dr. Zhu's research has been published in the Proceedings of National Academy of Sciences, Information System Research, INFORMS Journal on Computing, ACM Transactions, Naval Research Logistics, Annals of Statistics, Decision Support Systems, etc.