 http://dx.doi.org/10.17287/kmr.2021.50.3.665

What Leaders Really Do for Open Collaborations: Focusing on Open Source Software Development Projects*

개방형 협업 리더의 역할은 무엇인가?: 오픈소스 소프트웨어 프로젝트를 중심으로

Hyunmi Baek(First Author)
School of Media and Communication, Korea University
(*lotus1225@korea.ac.kr*)
Saerom Lee(Corresponding Author)
School of Business Administration, Kyungpook National University
(*saeromlee@knu.ac.kr*)
Sehwan Oh(Co-Author)
School of Business Administration, Kyungpook National University
(*sehwano@knu.ac.kr*)
Moonkyoung Jang(Co-Author)
Global IT department, Hannam university
(*mk.jang@hnu.kr*)

Rapid advancements in the field of information communication technology (ICT) have enabled people to collaborate through the Internet. This study focuses on open-source software development, which develops complex software or web services autonomously. To verify the antecedents of the participation of developers for successful open-source software development, we examine the relationship of the leadership roles of the project leader, such as knowledge contribution, interactions, and quality control with the participation of individual developers. For data collection, we develop a web crawler using the Python programming language and collect 518 repositories from GitHub, which is one of the leading platforms for developing open source software. We find that the participation of developers is closely associated with leaders who actively contribute knowledge, interact with the developers, and ensure strict quality control. Furthermore, the software development phase moderates the relationship between quality control and the participation of individual developers.

Key Words: Open collaboration, Open-source software, Innovation participation, Super leadership

Submission Date: 11. 09. 2020 Revised Date: (1st: 01. 16. 2021) Accepted Date: 01. 18. 2021

* This research is supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2019S1A3A2099973), and the MSIT(Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-0-01749-001) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation)

Ⅰ. Introduction

Open collaboration means "an online environment that supports the collective production of an artifact through a technologically mediated collaboration platform that presents a low barrier to entry and exit and supports the emergence of persistent but malleable social structure" (Forte & Lampe, 2013, p. 535). It has been noted that the outcomes of open collaborations are economically and functionally better than those in traditional firms (Andersen-Gott, Ghinea, & Bygstad, 2012; Faraj, Jarvenpaa, & Majchrzak, 2011; Hann, Roberts, & Slaughter, 2013; Melchor-Ferrer & Buendia-Carrillo, 2014; Rheingold, 2000; Tapscott & Williams, 2006). There are several examples of open collaboration, such as open-source software (OSS), Wikipedia, and crowdfunding. Open-source software development (OSSD), in particular, has a long history of collaboration and research streams on the success of these collaborations. As OSSD projects are developed by free contributions, it is important to enhance the developers' participation and performance to manage the project successfully. In this regard, previous research indicates that the success factors of OSSD performance are closely related to the role of leaders (Faraj, Kudaravalli, & Wasko, 2015; Giuri, Rullani, & Torrisi, 2008; Lerner & Tirole, 2001; Moon & Sproull, 2002; O'Mahony & Ferraro, 2004; Yoo & Alavi, 2004). With an increasing number of organizational forms targeting open collaboration, such as OSSD, there is an increasing need to look at what leadership needs for open collaboration with different structures and characteristics than a traditional organization. Hence, this research focuses on project leaders' roles that impact the participation of individual developers. We specifically examine the roles of a project leader in an OSSD project, such as knowledge contribution (the ratio of providing commits and the ratio of suggesting issues), interactions (the leader's out-degree centrality in the issue network), and quality control (the ratio of accepting code provided by other developers). Additionally, considering OSSD phases as a moderator, we try to verify the relationship between quality control by the project leader and the participation of individual developers.

This study tries to fill the gap left by previous studies regarding OSSD project leaders and suggests effective leadership roles in OSSD. It contributes to literature on the success of OSSD. First, we specifically suggest project leaders' roles from the perspective of super leadership. The stream of previous research on super leadership deals with the roles of leaders who distribute or share leaderships with employees in traditional firms (Manz & Sims, 1984; Jeong & Choi, 2015). Compared to traditional firms that have specified boundaries for employees, OSS is developed by autonomous developers who can participate in or leave the project anytime, anywhere. Since OSSD consists of free contributions made without any rewards or responsibilities, project leaders need to motivate developers to actively participate in projects and create a culture of collaboration (Hars & Ou, 2001; Lee, Baek, & Jahng, 2017; Markus, 2007). Therefore, the required capabilities of leaders in a virtual team such as OSSD may be different from those in traditional firms (Manz & Sims, 1984). Drawing on super leadership, this research suggests the importance of "leading by example", which implies motivating other developers to lead themselves and promote collaborations of individual developers. For leaders without the authority to assign responsibilities in OSSD projects, "leading by example" can be an especially effective strategy to encourage autonomous developers.

This study focuses on the leadership roles of project leaders, keeping changes in OSSD environments in mind. Most of the early research on OSSD leaders focused on the characteristics of emergent project leaders (Faraj et al., 2015; O'Mahony & Ferraro, 2004; Yoo & Alavi, 2004). Because famous projects such as Linux or UNIX need to appoint a promising developer as the project leader, it is important to verify the capabilities of other leaders. However, the evolution of web services for software development platforms such as GitHub and SourceForge has changed the OSSD environment and enabled developers to create new projects easily. As a result, there are significant changes in governance strategies or communication mechanisms of OSSD (Cosentino, Izquierdo, & Cabot, 2017). The research agenda related to the role of leaders in the success of a project is a crucial issue that has been recently dealt with (Lee, Baek, & Jahng, 2016).

This research aims to find antecedents in the participation of developers from the GitHub dataset. GitHub is a representative open-source software platform that began service in 2007, and it has more than 56 million developers and more than 100 million projects as of January 2021 (GitHub, 2021). GitHub provides social functionality that enables communication between developers and provides bulletin boards or wiki-type knowledge repositories within a project, making it easy for users to communicate and deliver code or information related to software development. In October 2018, Microsoft acquired GitHub for $7.5 billion, and it has been steadily moving toward a multinational open-source platform, with 75 percent of its developers from outside North America (GitHub, 2019).

In order to find the antecedents in the participation of developers, we focus on the project leaders' roles, such as knowledge contribution, interactions, and quality control in GitHub.

To examine the degree of activities of project leaders, we divide knowledge contribution into two parts: commits and issues. In GitHub, the activities of each developer are recorded as a 'commit' per project. GitHub also provides a feature called 'board', where developers can write issues in dealing with new ideas or problems in the project or suggest new code to project leaders. This feature helps in knowledge sharing. To measure the interactions of a leader, we use social network analysis (SNA) and calculate the out-degree centrality of a project leader based on the activities of providing comments on issues from other developers (Kwak, 2014; Scott, 2012; Wasserman & Faust, 1994). Finally, we define quality control as the ratio of accepting code suggested by other developers. For data analysis, we develop web crawlers using the Python programming language and collect data from a representative OSSD platform, namely GitHub. GitHub provides project pages called repositories for hosting the source code, commit records, issue boards, and information about projects (Dabbish et al., 2012). We gather data on 518 repositories and conduct a hierarchical regression analysis to test the hypotheses.

The rest of this paper is organized as follows: first, we provide previous research on OSSD and super leadership. Second, we present the hypotheses to verify the research model. Third, we explain the data collection and variables with methodological aspects of the study, followed by the results of the data analyses. The paper concludes by identifying the academic and practical implications and providing directions for further research.

II. Literature Review

2.1 Leaders of Open-Source Software Development Project

OSSD research has highlighted the role of leaders as a success factor for OSSD projects (Faraj et al., 2015; Giuri et al., 2008; Lerner & Tirole, 2001; Moon & Sproull, 2002; O'Mahony & Ferraro, 2004; Yoo & Alavi, 2004). Most of the research indicates that technical abilities, sociability, and interactions of leaders are contributing factors in the success of OSSD projects (Faraj et al., 2015; Kudaravalli, Faraj, & Johnson, 2017; Lee et al., 2017; Moon & Sproull, 2002; O'Mahony & Ferraro, 2004). As OSSD requires complex and professional knowledge, technical skill is stressed as an important characteristic of project leaders (Moon & Sproull, 2002). Similarly, leaders who contribute considerable knowledge to projects are known to display greater capability in managing projects (Faraj et al., 2015; Lee et al., 2017; O'Mahony & Ferraro, 2004). OSSD needs to collaborate in a computer-mediated communication (CMC) environment; however, there are significant differences between face-to-face (FTF) communication and CMC in terms of the existence of social context cues (Sproull & Kiesler, 1986). OSSD project leaders should play an active role in overcoming the disadvantages of CMC, which lacks non-verbal communication cues (Yoo & Alavi, 2004). Thus, it is important for sociable leaders to encourage other developers. In the context of OSSD, sociability is related to promoting developers, sharing an explicit purpose, or promoting collaborations (Faraj et al., 2015; Giuri et al., 2008; Lee et al., 2017; Lerner & Tirole, 2001; O'Mahony & Ferraro, 2004). Likewise, interactions of project leaders within a project are regarded as a core characteristic of the project's success (Faraj et al., 2015).

2.2 Super Leadership

Although there is extensive research available on leadership, research in virtual teams or online communities has a relatively short history (Avolio & Kahai, 2003; Cascio & Shurygailo, 2003; Kayworth & Leidner, 2001-2002; Zigurs, 2003). From the perspective of "leader as commander" (Gilbreth & Gilbreth, 1924; Taylor, 1911), open collaboration cannot assign responsibilities to others (Godfrey & Tu, 2000). Hence, different capabilities are required here compared to traditional firms (Manz & Sims, 1984). Manz & Sims (2001) introduced super leadership, which shares (Pearce & Conger, 2003) and distributes (Gronn, 2005) leadership and responsibilities of leaders among other developers. One of the characteristics of super leadership is "leading by example". Owing to the non-hierarchical structures of organizations, individual developers are expected to have leadership qualities and achieve high performance by themselves (Kiefer & Singe, 1999). For example, leaders can voluntarily teach the required performance, for example, how to actively participate in a project, how to strictly control the quality of job performance, and how much knowledge they can share with others. Specifically, super leadership focuses on leaders who voluntarily achieve the goal of a project and set an example (Manz & Sims, 2001). Consequently, leaders can help in enhancing individual developers' capabilities by personally guiding them in the right direction (Manz & Sims, 2001).

In the case of OSSD, developers voluntarily get involved in communities such as self-organizing systems. It is the nature of OSSD developers to have self-management and self-leadership skills in order to contribute knowledge or solve problems in projects (Scacchi, 2007). Considering that the gist of super leadership is sharing leadership with employees, OSSD requires sharing leadership with developers (Roberts, Hann, & Slaughter, 2006). Additionally, during the development of OSS, developers face various issues and often require professional knowledge. Project leaders cannot deal with every issue that occurs in a project; therefore, project leaders need to share leadership with developers for solving issues so that each developer has some responsibility in certain situations (Roberts et al., 2006).

Ⅲ. Hypotheses

Considering the knowledge contribution and technological capabilities of leaders, previous research has reported inconclusive issues related to the extent of knowledge leaders should contribute in projects. This impacts the individual developer's performance. An existing study on OSSD states that when the leader is not able to solve all problems, it is important to give a chance to other developers for ego-satisfaction by contributing to the project and helping advance their careers (Antikainen & Vaataja, 2010; Hars & Ou, 2001; Lerner & Triole, 2000). However, according to other research streams on roles of OSS leaders, technological skills and knowledge contribution are positively associated with innovation performance (Faraj et al., 2015; Lee et al., 2017; Moon & Sproull, 2002; O'Mahony & Ferraro, 2004). Considering the theoretical foundations of leadership, studies on super leadership show that the leader should take the initiative for others (Manz & Sims, 2001). Therefore, leaders with knowledge contribution in a project will be able to enhance an individual developer's OSSD activity. During the development of OSS, providing commits and suggesting issues for the project are directly related to the performance of the OSSD project. Regarding commit and issue suggestions in GitHub, we propose that increased knowledge contribution of a leader can enhance the participation of developers. Therefore, this study suggests the following hypothesis:

Hypothesis 1: Knowledge contribution of a project leader is positively associated with the participation of developers

Since OSSD develops complex software in a non-face-to-face environment, active collaborative communication among developers has a great effect on the success of a project (Faraj et al., 2015; Sonnentag et al., 1994) and existing organizations (Choi, Cho & Lee, 2020). Faraj et al. (2015) emphasized the importance of leaders' social capital formed through collaboration and communication with other developers in the projects. This study focused on providing feedback on ideas suggested by other developers, looking at it as interaction between leaders and developers. Previous research has asserted that feedback on the posted issues is important because it can improve upon ideas or solve problems (Moon & Sproull, 2002). In recently developed OSSD platforms, there is a built-in function for developers in the project to communicate with each other using boards and comments. The project leader understands the purpose or goals of the project well, and leaders can suggest the direction for a project or communicate with other developers through the feedback (Li, Tan, & Teo, 2012). Therefore, if developers suggest issues individually and the leader actively provides feedback on various issues and ideas rather than just concentrating on one or two messages or issues, it leads to a positive effect on other individual developers' OSSD activities. The SNA consists of the communication or activity among developers' nodes through a link (Kwak, 2014; Scott, 2012; Wasserman & Faust, 1994). In this study, each developer is marked as a node, and the behavior of providing comments and receiving comments is expressed as a link. Therefore, the out-degree centrality of a leader measures the structure of communication by focusing on feedback for other developers. The high out-degree centrality of a leader indicates that the leader provides comments to many developers. Therefore, the hypothesis is as follows:

Hypothesis 2: Interactions of a project leader are positively associated with the participation of developers

Since an OSSD project requires complex knowledge, developers apprehend the nature of the project to be developed before its development. Project leaders try to increase the quality of a project in an effort to improve its function and develop the software successfully (Hertel, Niedner, & Herrmann, 2003; Ho & Rai, 2017). In previous research, Ho and Rai (2017) deals with code acceptance as a signal of quality control. Developers can enhance the quality of the project code by playing the role of a code reviewer and reflecting the code change in the software. Signal theory is an economic theory to reduce information asymmetry between two sides to help make an effective decision (Boulding & Kirmani, 1993; Connelly et al., 2011; Spence, 1974). Based on the signal theory, the low acceptance rate of the code can give developers a signal that the project is strictly managed by reviewing the quality of the outcomes (Ho & Rai, 2017). Baldwin and Clark (2006) argued that a well-built code plays an important role in increasing the value of rewards, which encourages voluntary contributors to participate in or continuously contribute to the project. Marlow, Dabbish, and Herbsleb (2013) also insisted that wrong code contributed by developers wastes the time to be spent on the project and is a detour from the target result. Likewise, managing the project quality strictly encourages other developers to make contributions, thereby moving toward the goal more successfully. The hypothesis is as follows:

Hypothesis 3-1: Strict quality control by a project leader is positively associated with the participation of individual developers

As the OSSD process is different from a traditional software development cycle, Siau and Tian (2013) divided the process of OSSD into three phases: launch phases, the phase before the first release, and the phase between releases. Baysal and Malton (2007) also recognized that the development phase can be divided into two parts, namely, before releasing the software and after releasing it. Similarly, previous research has categorized the software development phase into whether software has been released or not. In line with previous research, we have adopted the same standard to classify the software development phases: before the first release and after the first release. While software planning is the gist of the roles before the release, the discussion on maintenance becomes the main task after the software is released. If the software reaches the after-release phase, the leader may intend to keep the already completed functions stable and in accordance with the original purpose rather than concentrate on a new idea or function developed by other developers. Hence, there will be stricter quality control over the newly suggested code. In the case of Linux, in the after-release phase, the bug is fixed or only the update of the previous stable release is considered (Godfrey & Tu, 2000). Since the project that has released the software keeps the functions stable, after software is released, strict project quality control will be more effective to developers' participation. Therefore, the hypothesis is as follows:

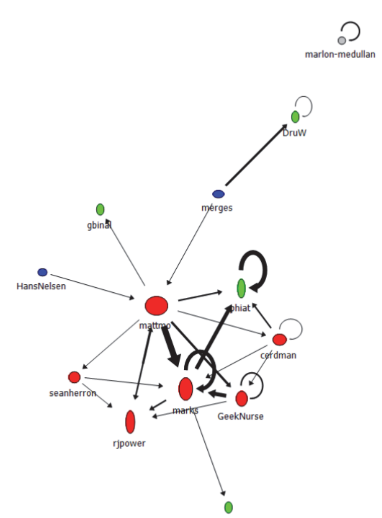
Hypothesis 3-2: Project maturity moderates the relationship between quality control by a project leader and the participation of developers.

Ⅳ. Research Methodology

4.1 Data Collection and Variables

We collected data on GitHub, a representative OSSD platform, through Web Crawler developed in the Python programming language. On GitHub, we targeted projects, called repositories, which had received more than five stars for a week since the creation of the repository among the repositories created from January 2014 to July 2014. The repositories owned by an organization were excluded from our analysis in order to look at the characteristics of the individual repository leader. Additionally, to control the duration of the repository, we targeted repositories that were operated for eight months (which is sufficient time to measure project results) after they were created. Finally, we formed an analysis of 518 repositories that had obtained more than three-closed issues. To infer the out-degree centrality of the leader, we analyzed the inter-developer comment network based on the issues proposed in the repository. We established the issue network of the repository by forming a link on the premise that developers who raised issues received comments from other developers. To measure the leaders' out-degree centrality, we conducted an SNA by creating a two-mode network between the repository's issues and developers. Then, we converted the two-mode network into a one-mode network, thus representing the relationship among repository developers. With this step, we were able to create 518 issue network structures for each repository. Figure 1 shows an example of the issue network structure among developers in a repository.

To study the behavior of developers' participation toward the success of the repository, we focused on the participation of developers (*Participation*), which is measured by the average commit number of each developer contributed in the target repository as a dependent variable. Before looking at the elements of leaders that affect performance of repository, we tried to look at the participation of developers as a dependent variable, a fundamental prerequisite for sustainability and performance of repository. Existing studies have used the number of commits contributed by each developer as the outcome of the project or the individual (Adams, Capiluppi, & Boldyreff, 2009; Crownston, Annabi, & Howison, 2003; Baek & Oh 2015). The leader's roles in a repository, as independent variables, were classified into three: the leader's knowledge contribution, interaction with other developers, and the project quality control. We measured knowledge contribution for commits and issue suggestion. Knowledge contribution regarding commits was measured by the ratio of commits contributed by the leader among the total commits in the repository (*Leader Commit*), and knowledge contribution with regard to issue suggestion was measured by the ratio of issues suggested by the leader among



〈Figure 1〉 The Issue Network Structure of the OSSD Repository (developer-developer)

〈Table 1〉 Variable Description

Variable		Measure
Dependent variable	*Participation*	The average commit number for each contributor in a repository
Independent variable	*LeaderCommit*	The ratio of commits by the leader among the total commits in a repository
	LeaderIssue	The ratio of issues and pull-requests suggested by the leader among the total issues and pull-requests in a repository
	LeaderCmtDeg	The out-degree centrality of the leader within a repository issue network, created by relationships between issues and developers
	LeaderMerge	The ratio of merged pull-requests among the total pull-requests in a repository
Moderating variable	*Release*	The number of releases of software in a repository

the total repository issues (*LeaderIssue*). Interaction was measured by the leader's out-degree centrality, derived from the result of the leader providing comments on issues raised by other developers in the issue network (*LeaderCmtDeg*). Quality control was measured by the ratio of commits merged by leader among the total commits proposed in the repository (*LeaderMerge*) (Ho & Rai, 2017). We used the number of releases of software in a repository (*Release*) as a moderating variable. The variables used in this research model are summarized in Table 1.

4.2 Analysis

The descriptive statistics of the variables are shown in Table 2. The average commit number in 518 repositories was 77.7, the contributor number was 5.4 on average, and the average commit number per contributor (*Participation*) as the dependent variable was 17.1. The average ratio of commits provided by the leader (*LeaderCommit*) was 77.4%. In most repositories, the leader tended to contribute to most commits. The ratio of issues raised by the leader (*LeaderIssue*) was 8.7% on average and the leader's out-degree centrality in the issue network (*LeaderCmtDeg*) was 1.67 on average. The average merge ratio (*LeaderMerge*) was 81.2%, while the average release count (*Release*) was 3.96 times.

Table 3 shows the correlation between variables in the research. The participation of developers (*Participation)* was positively related to the ratio of the leader's commits (*LeaderCommit*) (*correlation*=*0.2072*; *p-value* 〈 *0.01*). Additionally, the ratio of the leader's issue suggestion (*LeaderIssue*) (*correlation*= *0.3266*; *p-value* 〈 *0.01*) and the leader's out-

〈Table 2〉 Descriptive Statistics of Variables

Variable	N	Min	Max	Mean	Standard Deviation
LeaderCommit	518	0	0.993	0.774	0.218
LeaderIssue	518	0	0.931	0.087	0.171
LeaderCmtDeg	518	0	16	1.674	1.395
LeaderMerge	518	0	1	0.812	0.233
Release	518	0	93	3.956	9.072
Participation	518	2	141.5	17.081	18.887
Number of contributors	518	2	102	5.417	6.877
Number of commits	518	4	782	77.708	98.916

〈Table 3〉 Correlation Analysis

	1	2	3	4	5	6
1. LeaderCommit	1					
2. LeaderIssue	.1134***	1				
3. LeaderCmtDeg	-.0184	.0933**	1			
4. LeaderMerge	-.0646	.0944**	.0586	1		
5. Release	.0832*	.0988**	.2224***	-.1179***	1	
6. Participation	.2072***	.3266***	.2340***	-.0757*	.2727***	1

*N=518, *: p〈0.1, **: p〈 0.05, ***:p〈0.01*

degree centrality in the issue network (*LeaderCmtDeg*) (*correlation=0.2340, p-value* 〈 *0.01*) were positively correlated with the participation of developers (*Participation*), whereas the leader's merge ratio (*LeaderMerge*) was negatively correlated with the participation of developers (*Participation*) (*correlation=-0.0757, p-value* 〈 *0.1*).

We divided the repository into two types: the repository that released the software or the repository that did not. To compare these repositories, we conducted a t-test for the leader's role in a repository and the performance of the repository. Table 4 shows a comparison of descriptive statistics of sub-samples for a released repository and an unreleased repository. The project leader's knowledge contribution (*LeaderCommit*, *LeaderIssue*) was significantly higher in the repository that was released compared to their performance in the repository that was never released. These results suggest that the more knowledge is contributed by the project leader, the more software is released. In other words, the leader's knowledge contribution plays an important role in the successful operation of the OSSD project (Faraj et al., 2015; Lee et al., 2017).

In addition, the merge ratio (*LeaderMerge*) was lower in the repository that was released compared to the repository that was not. In other words, as the software development phase matured, the project quality was more strictly controlled. In the after-release phase, the leader tried to keep the functions in line with the original purpose (Godfrey & Tu, 2000). In this regard, leaders strictly controlled the quality of the new code as it contained new features or fresh ideas. However, there was no change in the leader's interaction with other developers (*LeaderCmtDeg*), which was measured by the out-degree centrality of the leader in the issue network. Even when the release was made, there was no significant change in activities such as the leader's comments on issues suggested by other developers.

In terms of the outcomes of the repository, in the released repository, the sum of the commit number and the average commit number per contributor (*Participation*) was significantly higher, but the sum of the contributor number had nothing to do with whether the repository was released or not. This means that after the release, the developer does not actively participate in the repository, and the average commit number of the existing contributors who have already participated (*Participation*) tends to increase. The result of commit numbers per contributor corresponds with those by Raymond (1999), who identified early and frequent releases to keep the contributors interested and rewarded. In previous research, more developers joined the

〈Table 4〉 Descriptive Statistics and Comparison between Sub-sample Means

Variable	Released Repository (n=261) M(SD)	Unreleased Repository (n=257) M(SD)	P-value
LeaderCommit	0.794 (0.014)	0.753 (0.013)	0.0296*
LeaderIssue	0.107 (0.011)	0.066 (0.010)	0.0067**
LeaderCmtDeg	1.771 (0.083)	1.575 (0.090)	0.1093
LeaderMerge	0.781 (0.016)	0.843 (0.013)	0.0025**
Participation	19.061 (1.188)	15.070 (1.147)	0.0161*
Number of contributors	5.372 (0.325)	5.463 (0.512)	0.8800
Number of commits	91.452 (6.950)	63.751 (5.056)	0.0014**

Notes: Using the t-test
* $p < 0.05$, ** $p < 0.01$

project after the release (Hahn, Moon, & Zhang, 2008; Lerner & Tirole 2002), but our study shows that the number of developers who join the repository after the release was not changed statistically. Before the release of software, various developers participate in the OSSD repository, but after the release, they focus on conservative activities, such as maintaining the released software or upgrading functions. It can be said that there are no significant newcomers and development is deeper when centered on contributions made before a release.

Ⅴ. Results

We performed hierarchical regression using Stata 14 to analyze the hypotheses. Baron and Kenny (1986)'s verification method was used to analyze the moderating effect. The Durbin-Watson statistic was 2.055, which suggested that there was no autocorrelation. We further examined the variance inflation factor (VIF) of each independent variable and the interaction terms. The average VIF was 6.08, with the largest VIF at 16.11. When the largest VIF value was greater than 10, it indicated the existence of a potential multicollinearity problem (Alin, 2010). To address this, mean centering was used for independent variables and interaction terms.

In Model 1, we considered only independent variables: the leader's commit ratio, issue ratio, out-degree centrality in the issue network, and the merge ratio. The analysis result showed the impact of the leader's roles on the participation of individual developers. It was noted that the higher the ratio of the leader's commit (*LeaderCommit*), issue suggestion (*LeaderIssue*), and the leader's out-degree centrality (*LeaderCmtDeg*), the better the participation of developers (*Participation*). However, as the merge ratio (*LeaderMerge*) decreased, the participation of developers (*Participation*) increased. The explanation power of Model 1 was 19.1%. In Model 2, we added the release count as a moderating variable. The explanation power of Model 2 was 22.2%, showing an increase of 3.1% from Model 1. The release count (*Release*) is positively related to the participation of developers (*Participation*). In Model 3, to verify the moderating effect of the release count, the interaction terms of the merge ratio and release count were added. As this had a negative impact on the number of commits per contributor, it was observed that as the release count increased, the merge ratio's negative impact on the participation of developers increased as well. This suggests that the release count moderated the impact of the merge ratio on the participation of individual developers. As a result of analyzing the moderating effect of the release count

〈Table 5〉 Analysis Results of Hierarchical Regression Model

Variable	Model 1		Model 2		Model 3	
	Coefficient	SE	Coefficient	SE	Coefficient	SE
LeaderCommit	14.780**	3.472	13.680**	3.417	13.884**	3.382
LeaderIssue	32.773**	4.452	31.098**	4.386	32.396**	4.357
LeaderCmtDeg	2.921**	.541	2.365**	.545	2.306**	.540
LeaderMerge	-8.555**	3.252	-6.555*	3.223	-6.255	3.191
Release			.382**	.085	.290**	.088
Release × LeaderMerge					-1.431**	.416
R^2	.191		.222		.239	
Adjusted R^2	.185		.214		.230	
△F	6.919**		20.355**		11.823**	
N	518		518		518	

Note: *$p < .05$, **$p < .01$, Dependent variable: Participation

(*Release*) on the impact of the merge ratio (*LeaderMerge*) on the participation of developers (*Participation*), the explanation power increased by a significant 1.7%. The outputs from these hierarchical regression models are included in Table 5.

For the robustness check, we further analyzed the moderating effect of the release (released repository versus unreleased repository) on the relationships between the leader's roles in the repository (independent variables) and the participation of developers (dependent variable). For this, Chin's t-value was calculated based on multiple regression analysis results in released and unreleased repository groups. We verified the existence of moderating effects depending on the repository release using equation (1) suggested by Chin (2004).

$$t = \frac{Path_1 - Path_2}{\sqrt{\frac{(n_1-1)^2}{(n_1+n_2-2)} * SE_1^2 + \frac{(n_2-1)^2}{(n_1+n_2-2)} * SE_2^2} * \sqrt{\frac{1}{n_1}+\frac{1}{n_2}}} \quad (1)$$

$Path_i$: *coefficient of path i*
n_i : *sample size of path i*
SE_i : *standard error of path i*

Table 6 contains the moderating effects of the release based on Chin's t-value. According to the result of Chin's t-test, only the relationship between the merge ratio and the four leader's roles is taken into account in this study model. The participation of developers is affected by whether the repository is released or not. The impact of the leader's commit ratio (*LeaderCommit*), issue ratio (*LeaderIssue*), and out-degree centrality (*LeaderCmtDeg*) on the participation of developers (*Participation*) did not change depending on the release of

〈Table 6〉 Analysis Results of Moderating Effects

Variable	Released Repository		Unreleased Repository		Chin's t-value	Moderating Effect
	Coefficient	SE	Coefficient	SE		
LeaderCommit	13.778**	4.797	16.562**	5.044	-0.40	Not supported
LeaderIssue	32.714**	6.008	31.339**	6.736	0.15	Not supported
LeaderCmtDeg	3.893**	.797	2.013**	.739	1.73	Not supported
LeaderMerge	-14.666**	4.218	1.343	5.199	-2.40*	Supported
R^2	.233		.154			
Adjusted R^2	.221		.141			
N	261		257			

Note: *$p < .05$, **$p < .01$, *Dependent variable: Participation*

software. However, the impact of the merge ratio (*LeaderMerge*) on the participation of developers (*Participation*) changed depending on the release of the software. In the released repository, the stricter the quality control, the better the developer's participation. On the other hand, quality control is not related to the developer's participation in the unreleased repository, supporting Hypothesis 3-2.

Ⅵ. Discussion and Conclusion

This study analyzed the factors that affect the participation of developers in the OSSD project, focusing on project leaders' roles. As a result, we observed that the higher the ratio of commits and issue suggestions by project leaders, the better the participation of developers in the project. Existing research states that the contribution of other participants may be low due to the diffusion of responsibilities when one or two people contribute too much in open communities (Darley & Latane, 1968; Yan & Jian, 2017). However, this study found that the leaders' knowledge contribution has a positive effect on individual developers' participation. In addition, the more the leader comments on issues raised by other developers, the better the developer's participation. This is consistent with the previous study, which noted that performance is greater when leaders act as mentors in the CMC environment (Anvik & Murphy, 2011; Canfora et al., 2012; Dagenais et al., 2010). Furthermore, developer participation in innovation increases when managers strictly monitor the quality of projects.

We also investigated the leader's roles in the OSSD project and the impact of these roles on the developers' participation depending on the release of software. As the phase of software development matures, the leader's knowledge contribution (i.e., the ratio of commits and issue suggestions) increases as well, while quality control of the repository becomes stricter. In terms of the outcome of the repository, the total number of commits and the commit number per contributor were significantly higher in released repositories than in unreleased repositories. The number of contributors is not related to whether the repository was released or not. The impact of project quality control on the developer's innovation participation varies depending on the software development phase, whereas the impact of the leader's ratio of providing commits, suggesting issues, and facilitating interaction with other developers is consistent regardless of the software phase. The greater the number of releases, the higher the impact of the project quality control on the developer's performance.

There are several limitations of this research. First, we do not consider changes in each project with time series data. Although this study has attempted to analyze the phase of software development depending on the release of the repositories on GitHub, there is no way to investigate the roles of the leader and the participation of developers from the time the repository was created. Furthermore, the findings of this study could be generalized by examining the characteristics of open collaboration leaders who can elicit voluntary participation from various open collaboration platforms other than GitHub. Another limitation is measuring the participation of developers as a dependent variable. The participation of developers can be measured in various ways, such as the number of issue suggestions, interactions with other developers, and quality of code. However, we consider only the number of commits for the participation of developers. To overcome these limitations, future research may conduct various methodologies such as collecting survey data or time series data. This study could not analyze the performance of repository as a limitation of data acquisition. If data is available to measure performance, such as the number of downloads of software developed in repository, the role of leaders in influencing collaboration performance could be analyzed.

Despite these limitations, this research makes some multifaceted contributions. First, as a contribution to theory, it is academically meaningful to investigate an effective leader's governance strategy in an open collaboration based on the super leadership theory. We note the role of leaders in managing OSSD by evaluating leadership as "leading by example", which was rarely dealt with in depth in the field of governance strategy or leadership in traditional firms. In addition, we investigate the leader's governance strategy depending on the phase of software development.

Second, as a contribution to methodology, we collected and analyzed data on the leader's governance characteristics and participants' activity for the actual 518 repositories in the representative open source project platform, GitHub. Based on data about the issues of the 518 target repositories, 518 issue networks among the developers in each repository were constructed through social network analysis, and the owner characteristics of interaction with other participants were derived. We were able to analyze factors that affect developers' participation with more variables through this process.

Finally, the current research contributes to practice. In the case of collaborations within traditional firms, since the leader has special authority, it is possible to manage collaborations using authority. However, since the leader does not have special authority in an open collaboration, it is important to have the foresight to draw voluntary collaborations. This study explored the leader's governance characteristics to encourage voluntary and continuous participation in an open collaboration. The findings of this study can be used as guidelines to design an open collaborative platform effectively. To encourage other participants, it is important to design the platform to reveal the leader's exemplary displays of knowledge contribution, interaction with other participants, and signals that the leader is constantly managing the quality of the collaboration. A strategy is needed to index and present the characteristics of the leaders derived from this study so that developers who feel uneasy about the sustainability of the repository can decide whether to participate.

References

Adams, P., Capiluppi, A., and C. Boldyreff(2009), "Coordination and Productivity Issues in Free Software: The Role of Brooks' Law," in proceedings of IEEE International Conference on Software Maintenance, pp.319-328.

Alin, A.(2010), *Multicollinearity*, Wiley Interdisciplinary Reviews: Computational Statistics, 2(3), pp.370-374.

Andersen-Gott, M., Ghinea, G., and B. Bygstad (2012), Why Do Commercial Companies Contribute to Open Source Software?," *International Journal of Information Management,* 32(2), pp.106-117.

Antikainen, M., and H. Vaataja(2010), "Rewarding in Open Innovation Communities - How to Motivate Members," *International Journal of Entrepreneurship and Innovation Management,* 11(4), pp.440-456.

Anvik J., and G. Murphy(2011), "Reducing the Effort of Bug Report Triage: Recommenders for Development-Oriented Decisions," *ACM Transactions on Software Engineering and Methodology*, 20(3), pp.10.

Avolio, B., and S. Kahai(2003), "Adding the "E" to E-Leadership: How It May Impact Your Leadership," *Organizational Dynamics,* 31 (4), pp.325-338.

Baldwin, C., and K. Clark(2006), "The Architecture of Participation: Does Code Architecture Mitigate Free Riding in the Open Source Development Model?," *Management Science,* 52(7), pp.1116-1127.

Baron, R., and D. Kenny(1986), "The Moderator-Mediator Variable Distinction in Social Psychological Research: Conceptual, Strategic, and Statistical Considerations," *Journal of Personality and Social Psychology,* 51(6), pp.1173-1182.

Baysal, O., and A. Malton(2007), "Correlating Social Interactions to Release History During Software Evolution," in proceedings of Fourth International Workshop on Mining Software Repositories.

Baek, H., and Oh, S.(2015), "Identifying the Network Characteristics of Contributors that Affect Performance in Open Collaboration: Focusing on the Github Open Source," *Korea Institute of Science and Technology Information,* 20 (1), pp.23-43.

Boulding W, and A. Kirmani(1993), "A Consumer-Side Experimental Examination of Signaling Theory: Do Consumers Perceive Warranties as Signals of Quality?," *Journal of Consumer Research,* 20(1), pp.111-123.

Canfora, G., Di Penta, M., Oliveto, R., and S. Panichella(2012), "Who is Going to Mentor Newcomers in Open Source Projects?," in proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering, 44.

Cascio, W. F., and S. Shurygailo(2003), "E-Leadership and Virtual Teams," *Organizational Dynamics,* 31(4), pp. 362-376.

Chin, W.(2004), "Frequently Asked Questions—Partial Least Squares & PLSgraph," http://Disc-Nt.Cba.Uh.Edu/Chin/Plsfaq.Htm (retrieved 9 January 2018).

Choi, Y., Cho, B., and Lee, E.(2020), "The Effect of Empowering Leadership on Follower's Promotive Voice Behavior: Mediating Effect of Psychological Safety and Moderated Mediating Effect of Coworker's Acquiescent Silence," *Korean Management Review,* 49(6), pp.1549-1573.

Connelly B., Certo St, Ireland Rd, Cr. Reutzel(2011), "Signaling Theory: A Review and Assessment," *Journal of Management,* 37(1), pp.39-67.

Cosentino, V., Izquierdo, J. L. C., and J. Cabot(2017), "A Systematic Mapping Study of Software Development with Github," *IEEE Access,* 5, pp.7173-7192.

Crowston, K., Annabi, H., and J. Howison(2003), "Defining Open Source Software Project Success," in proceedings of the International Conference of Information Systems, 2003, 28.

Dabbish, L., Stuart, C., Tsay, J., and J. Herbsleb (2012), "Social Coding in Github: Transparency and Collaboration in an Open Software Repository," in proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, pp.1277-1286.

Dagenais, H. Ossher, R. K. E. Bellamy, M. P. Robillard, and J. De Vries(2010), "Moving into a New Software Project Landscape," proceeding of the 32nd Acm/Ieee International Conference on Software Engineering - Volume 1, ICSE 2010, Cape Town, South Africa, 1-8 May 2010, pp. 275-284.

Darley, J. M., and B. Latane(1968), "Bystander Intervention in Emergencies: Diffusion of Responsibility," *Journal of Personality and*

Social Psychology, 8(4p1), pp. 377-383.

Faraj, S., Jarvenpaa, S. L., and A. Majchrzak (2011), "Knowledge Collaboration in Online Communities," *Organization Science,* 22(5), pp.1224-1239.

Faraj, S., Kudaravalli, S., and M. Wasko(2015), "Leading Collaboration in Online Communities," *MIS Quarterly,* 39(2), pp.393-412.

Forte, A., and C. Lampe(2013), "Defining, Understanding, and Supporting Open Collaboration: Lessons From the Literature," *American Behavioral Scientist,* 57(5), pp.535-547.

Gilbreth, F. B., and L. M. Gilbreth(1924), "Classifying the Elements of Work," *Management and Administration,* 8(2), pp. 151-154.

Github(2019), The State of the Octoverse, https://octoverse.github.com(Retrieved April 25, 2020).

Github(2021), https://github.com (Retrieved January 11, 2021).

Giuri, P., Rullani, F., and S. Torrisi(2008), "Explaining Leadership in Virtual Teams: The Case of Open Source Software," *Information Economics and Policy,* 20(4), pp.305-315.

Godfrey, M. and Q. Tu(2000), "Evolution in Open Source Software: A Case Study," in proceedings of Software Maintenance.

Gronn, P,(2005), *Distributed Organizational Leadership*. Greenwich, Ca: Information Age Publishing.

Hann, I. H., Roberts, J. A., and S. A. Slaughter (2013), "All are not Equal: An Examination of the Economic Returns to Different Forms of Participation in Open Source Software Communities," *Information Systems Research,* 24(3), pp.520-538.

Hahn, J., Moon, J. Y., and C. Zhang(2008), "Emergence of New Project Teams from Open Source Software Developer Networks: Impact of Prior Collaboration Ties," *Information Systems Research,* 19(3), pp. 369-391.

Hars, A. And S. Ou(2001), "Working for Free? Motivations of Participating in Open Source Projects," in proceedings of the 34th Annual Hawaii International Conference on System Sciences.

Hertel, G., Niedner, S., and S. Herrmann(2003), "Motivation of Software Developers in Open Source Projects: An Internet-Based Survey of Contributors to the Linux Kernel," *Research Policy,* 32(7), pp.1159-1177.

Ho, S. Y., and A. Rai(2017), "Continued Voluntary Participation Intention in Firm-Participating Open Source Software Projects," *Information Systems Research,* 28(3), pp.451-679.

Jeong, G. J., and Choi, S. B,(2015), "Exploring the Relationship Between Super-Leadership and Innovation Behavior : Mediating Effects of Followership and Positive Psychological Capital," *Journal of Business Research,* 30 (2), pp.1-30.

Kayworth, T., and D. E. Leidner(2001-2002), "Leadership Effectiveness in Global Virtual Teams," *Journal of Management Information Systems,* 18(3), pp 7-40.

Kiefer, F., and P. M. Senge(1999), "Metanoic Organizations in the Transition to a Sustainable Society," *Reflections: The Sol Journal,* 1(1), pp.25-36.

Kudaravalli, S., Faraj, S., and S. L. Johnson(2017), A Configural Approach to Coordinating Expertise in Software Development Teams," *MIS Quarterly,* 41(1), pp.43-64.

Kwak, G.(2014), *Social Network Analysis*, Cheongram.

Lee, S., Baek, H., and Jahng, J.(2017), "Role of Project Owner in OSS Project: Based on Impression Formation and Social Capital Theory," *Journal of Society for E-Business Studies,* 21(2), pp.23-46.

Lee, S., Baek, H., and J. Jahng(2017), "Governance Strategies for Open Collaboration: Focusing on Resource Allocation in Open Source Software Development Organizations," *International Journal of Information Management,* 37(5), pp,431-437.

Lerner, J. and J. Tirole(2001), "The Open Source Movement: Key Research Questions," *European Economic Review,* 45(4), pp. 819-826.

Lerner, J., and J. Tirole(2002), "Some Simple Economics of Open Source," *Journal of Industrial Economics,* 50(2) pp.197-234.

Li, Y., Tan, C. H., and H. H. Teo(2012), "Leadership Characteristics and Developers' Motivation in Open Source Software Development," *Information & Management,* 49(5), pp.257-267.

Marlow, J., Dabbish, L., and J. Herbsleb(2013), "Impression Formation in Online Peer Production: Activity Traces and Personal Profiles in Github," in proceedings of the 2013 Conference on Computer Supported Cooperative Work, pp.117-128.

Manz, C., and H. Sims(1984), "Searching for the 'Unleader': Organizational Member Views on Leading Self-Managed Groups," *Human Relations,* 37(5), pp.409-424.

Manz, C., and H. Sims(2001), *The New Superleadership: Leading Others to Lead Themselves.* Berrett-Koehler Publishers.

Markus, M.(2007), "The Governance of Free/Open Source Software Project: Monolithic, Multi-dimensional, or Configurational?," *Journal of Manage Governance,* 11(2), pp. 151-163.

Melchor-Ferrer, E., and D. Buendía-Carrillo(2014), "Financial Information Management for University Departments, Using Open-Source Software," *International Journal of Information Management,* 34(2), pp. 191-199.

Moon, J., and L. Sproull(2002), "Essence of Distributed Work: The Case of the Linux Kernel," *Distributed Work*, London, England, pp. 381-404.

O'Mahony, S., and F. Ferraro(2004), "Hacking Alone? The Effects of Online and Offline Participation on Open Source Community Leadership," *Working Paper*.

Pearce, C., and J. Conger(2003), "All those Years Ago: The Historical Underpinnings of Shared Leadership. In C. L. Pearce, & J. A. Conger (Eds.)," *Shared Leadership: Reframing the Hows and Whys of Leadership* (pp.1-18). Thousand Oaks, Ca: Sage.

Raymond, E(1999), "The Cathedral and the Bazaar," *Knowledge*, 12(3), pp. 23-49.

Rheingold, H(2000), *The Virtual Community: Homesteading on the Electronic Frontier*. Cambridge, Ma: Mit Press.

Roberts, J., Hann, I., and S. Slaughter(2006), "Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects," *Management Science,* 52 (7), pp. 984-999.

Scacchi, W(2007), "Free/Open Source Software Development," in proceeding of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software

Engineering, pp.459-468.

Scott, J.(2012), *Social Network Analysis*. Sage.

Siau, K., and Y. Tian(2013), "Open Source Software Development Process Model: A Grounded Theory Approach," *Journal of Global Information Management,* 21(4), pp. 103-120.

Sonnentag, S., Frese, M., Stolte, W., Heinbokel, T., and F. Brodbeck(1994), "Goal Orientation of Team Leaders: Its Effects on Performance and Group Interaction in Software Development Projects," *European Journal of Work and Organizational Psychology,* 4(2), pp. 153-168.

Spence, M.(1974), *Market Signaling: Informational Transfer in Hiring and Related Screening Processes*, Cambridge, Ma: Harvard University Press.

Sproull, L., and S. Kiesler(1986), "Reducing Social Context Cues: Electronic Mail in Organizational Communication," *Management Science,* 32(11), pp. 1492-1512.

Tapscott, D., and A. D. Williams(2006), *Wikinomics: How Mass Collaboration Changes Everything*, New York: Portfolio.

Taylor, F.(1911), *The Principles of Scientific Management*, New York: Harper & Brothers.

Wasserman, S., and K. Faust(1994), *Social Network Analysis: Methods and Applications*, Cambridge University Press.

Yan, B., and L. Jian(2017), "Beyond Reciprocity: The Bystander Effect of Knowledge Response in Online Knowledge Communities," *Computers in Human Behavior,* 76, pp. 9-18.

Yoo, Y. and M. Alavi(2004), "Emergent Leadership in Virtual Teams: What Do Emergent Leaders Do?," *Information and Organization,* 14(1), pp. 27-58.

Zigurs, I.(2003), "Leadership in Virtual Teams: Oxymoron or Opportunity?," *Organizational Dynamics,* 31(4), pp. 339-351.

- The author Hyunmi Baek is an associate professor in the School of Media and Communication, Korea University. She received her BS in chemical engineering from the Pohang University of Science and Technology and her MS in IT management from the Information and Communications University. She received her PhD in management information systems from Seoul National University. From 2003 to 2013, she worked for Electronics and Telecommunications Research Institute. From 2013 to 2018, she worked for Hanyang University. Her primary research interests are electronic word-of-mouth in social media, media management, information adoption, and open collaboration.

- The author Saerom Lee has been with the School of Business Administration, Kyungpook National University, Daegu, Rep. of Korea, where she is now an assistant professor, since 2018. She received her BS degree in International Trade from Pusan National University, in 2010, and her PhD degree in management information systems from Seoul National University, Seoul, Rep. of Korea, in 2016. Her main research interests are open collaboration in open source software developers and cyber bullying behaviors focusing on sexual harassment.

- The author Sehwan Oh is an Associate Professor at the School of Business Administration, Kyungpook National University. He earned his PhD Degree in Management Information Systems from the College of Business Administration, Seoul National University, MS Degree in Information Technologies – eBusiness Technology (MSIT) from the School of Computer Science, Carnegie Mellon University, and BA Degree in Economics from the College of Social Sciences, Seoul National University. He worked for Korea International Trade Association (KITA) as a senior researcher. His current research interests include sharing economy, social media, electronic word-of-mouth, and ICT use in international trade.

- The author Moonkyoung Jang is an assistant professor at Global IT Business, Hannam university. She received her Ph.D. Degree in Management Information Systems from the College of Business Administration, Seoul National University. She worked as an ERP programmer for Lotte Data Communication Company and Tyco International, and worked as a researcher at Nanyang Technological University and Korea University. Her research interests include digital contents platform, eWOM, open collaboration, and media management.