Received: 25 September 2018 | Revised: 9 July 2019 | Accepted: 14 August 2019

DOI: 10.4218/etrij.2018-0536

ORIGINAL ARTICLE

ETRI Journal WILEY

The role of openness in open collaboration: A focus on open-source software development projects

Saerom Lee[1] | **Hyunmi Baek**[2] | **Sehwan Oh**[1]

[1]School of Business Administration, Kyungpook National University, Daegu, Rep. of Korea

[2]School of Media and Communication, Korea University, Seoul, Rep. of Korea

Correspondence
Hyunmi Baek, School of Media and Communication, Korea University, Seoul, Rep. of Korea
Email: lotus1225@korea.ac.kr

Funding information
This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2019S1A3A2099973)

Open-source software development projects are well suited for exploring new ideas and acquiring knowledge from developers outside of the project. In this paper, we examine the impact of external developers on innovation in open-source software development from the perspective of organizational learning theory. We examine the roles of external and internal developers, who "explore" and "exploit," respectively, on the innovation performance of 17 691 open-source software development projects whose data is stored in the GitHub platform. The results indicate that a multifaceted strategy, in which the exploitation successfully supports the exploration, is most effective for their success. The results also indicate that the role of exploration decreases after the release of the software.

KEYWORDS
exploitation, exploration, innovation management, open-source software, software development

1 | INTRODUCTION

A company's innovation directly affects its survival [1,2]. In recent years, with the development of information and communication technology, open collaboration via the Internet has been used as a strategic approach for innovation [3]. In open collaboration, developers work together for a specific purpose through the Internet, utilizing external resources to create innovative products [3].

Because open collaborations via the Internet have a high probability of failure [4,5], their method of operation should be discussed often [6–8]. Open-source software development (OSSD) is an example of open collaboration. In this paper, we investigate the factors affecting the performance of open collaborations with an emphasis on OSSD projects. With developments in OSSD platform environments, developers are easily able to create the software involved in the collaboration. The number of projects whose data is stored in Github (https://github.com) or SourceForge (https://sourceforge.net) has increased rapidly, because many people can open projects in the platform. However, OSS projects depend on the participation of autonomous developers, and approximately 74.2% of open projects have only two developers [4]. The results of our study provide a governance strategy for OSSD organizations to allocate external and internal developers in running projects.

The number of developers participating in a project directly affects the success of the project [9]. It is commonly assumed that successful OSSD projects are those involving more, rather than fewer, developers. Since open collaboration projects possess an open mechanism for adopting external resources and knowledge, in our study we wished to thoroughly investigate the role of external developers in the success of the project. Organizational learning theory was adopted as a theoretical lens through which to interpret the effects of external and internal developers on OSSD project performance, the latter of which was measured by the number of commits.

Organizational learning theory focuses on the activities of knowledge creation companies that use both internal and external resources to identify the factors influencing the companies' innovations [10]. March [11] suggested that knowledge creation activities can be divided into two categories: exploration (exploring external resources) and exploitation (refining the internal resources of the firm). To discuss the issue of OSSD project openness, we divide developers into two categories: those who explore and those who exploit. Our work is based on that of March [11].

An OSSD project is based on the collaboration of a number of independent developers whose innovations are based on the exploration of external resources. We refer to the developers' knowledge creation as *exploration*.

With the evolution of OSSD platforms, multiple developers can organize a team ("organization" in GitHub) and collaborate with other organization members to develop software [6]. Because organization members' innovation is based on existing internal resources, their use of these resources is referred to as *exploitation*.

Focusing on the roles of exploration and exploitation in OSSD projects, we present two research questions: *What are the impacts of exploration and exploitation on the results of an OSSD project? How does the phase of the OSSD impact the effect of exploration on a project's results?* To answer these questions, we defined exploration (exploitation) to be the percentage of external (internal) developers participating in an OSSD project from among all the developers participating in the project.

For data analysis, we developed a web crawler using the Python programming language and collected data from a typical OSSD platform, namely, GitHub. GitHub provides project pages called repositories for hosting source code, commit records, issue boards, and other project information [12]. We gathered data from 17 691 repositories that are managed by organizations (rather than individuals) and conducted a hierarchical regression analysis to test our hypotheses.

The remainder of this paper is organized as follows. In the second section, previous theoretical studies, including OSSD governance and resource allocation, are reviewed. In Section 3, a research model and relevant hypotheses to verify the model are presented. Section 4 describes the data obtained from GitHub for analysis. Section 5 presents the analytical results. Finally, Section 6 presents the conclusions of this study, as well as the research limitations.

2 | LITERATURE REVIEW

2.1 | Open-source software development projects

Open-source software (OSS), which is the product of open collaboration, is defined as "software that allows for the modification of source code, is freely distributed, is technologically neutral, and grants free subsidiary licensing rights" [13]. Since the 1990s, OSSD projects have been conducted by developers for various purposes in Internet communities. To provide a convenient environment for OSS developers, OSS-focused web services have emerged, for example, Google Code, SourceForge, and GitHub. GitHub, which has provided services since 2008, supports the work of around 28 million developers and has 85 million repositories as of September 17, 2018.

To support developers who often collaborate with each other, GitHub provides a feature called an "organization" [14]. An organization can create and manage multiple projects established by organization members. Figure 1 shows Google's GitHub organization page.

GitHub enables members who often collaborate to share their results with other members of the organization and the characteristics and development cultures of the organization members. The organization members also collaborate with external developers who are not a part of the organization. By providing opportunities for external developers to participate in the project, collaboration allows the internal developers to adopt external knowledge. Thus, one of the roles of external developers is to provide ideas from outside the project to help the project succeed. The internal members focus both on creating new code individually and on managing the code submitted by the external developers.

2.2 | Organizational learning theory

Organizational learning is defined as "the experiential production and reproduction of organizational rules, leading to behavioral stability or behavioral changes" [15,16]. Typically, learning activities are divided into two categories: exploration activities and exploitation activities [17–23]. According to March [11], exploitation is considered to be a refinement and extension of a firm's existing internal knowledge. Exploitation involves creating output in the technological domain in which the companies apply patents [24].

On the other hand, exploration focuses on investigating external resources for knowledge creation. Such exploration-oriented activities assist a firm in developing new ideas and acquiring the capabilities necessary for survival and long-term success. However, the results are uncertain and often slow to manifest.

As March [11] did not provide explicit definitions for exploration and exploitation, researchers have interpreted these concepts in various ways. Some researchers regard exploration and exploitation as innovation processes, for example, organizational learning activities, behaviors, investments, or strategies [10,22,25–29]. To other researchers, these concepts represent the outcome of the innovation process [17,21,26].

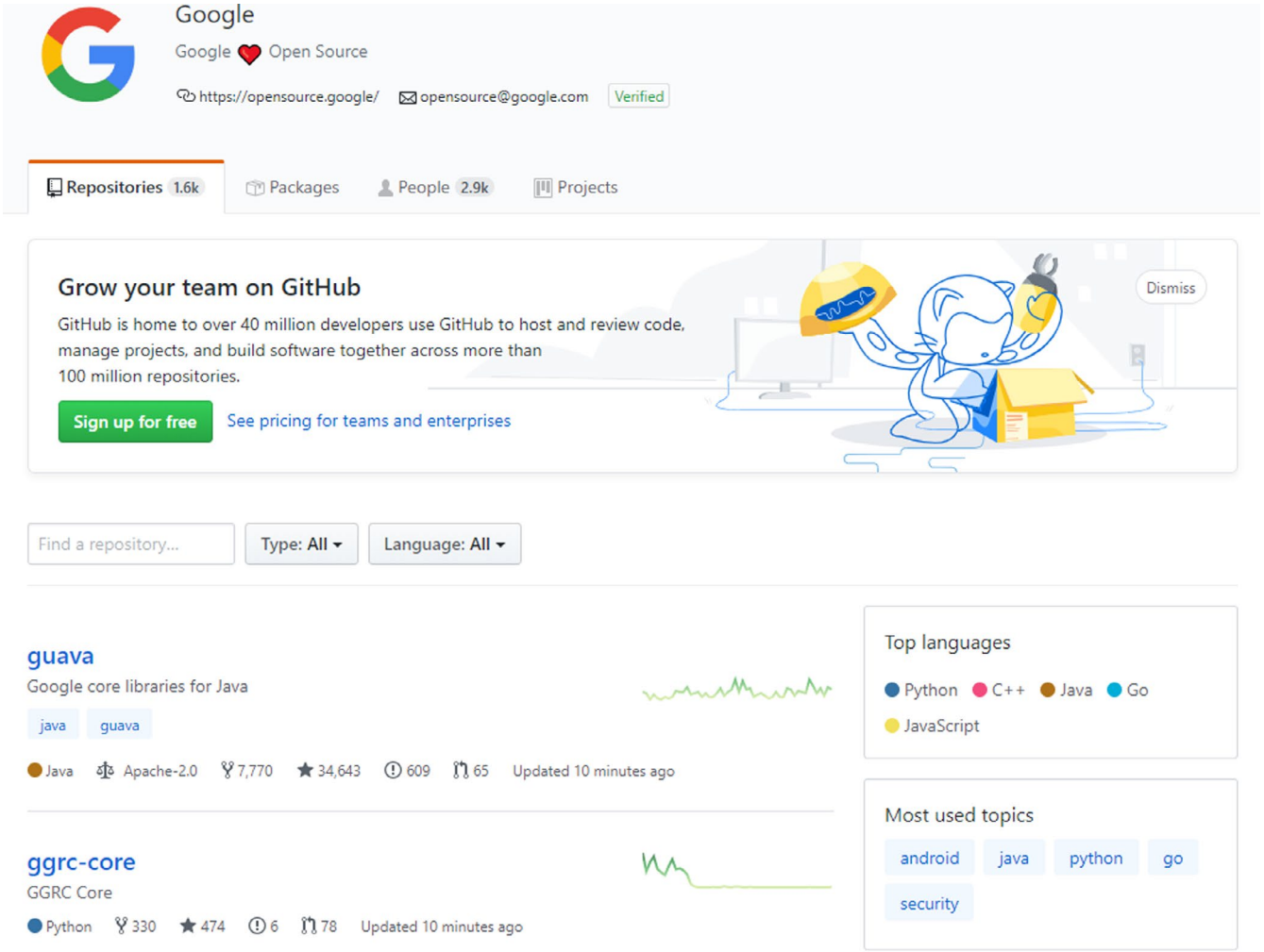

FIGURE 1 Example of a GitHub organization ("Google" organization)

In open collaboration, the project's outcome is affected by the composition of the organization. Therefore, we studied the factors that contribute to an OSSD project's outcome using the concepts of exploration and exploitation. Hence, it was necessary to develop a method for measuring exploration and exploitation. Perretti and Negro [30] considered exploitation to be coordination with prior collaborators and exploration to be coordination with new developers. They asserted that the role of newcomers is to search for and provide new ideas. In this vein, newcomers can expand the firm's intellectual boundaries and enable access to new external resources [31].

March [11] proposed that existing members have already contributed their ideas to the organization. In contrast, newcomers might not know as much about the organization as the existing members, but they are able to contribute new intellectual resources.

Although previous studies have also considered organizational learning theory to be a theoretical lens through which to view OSSD projects [32], they consider OSS to be the result of a collective endeavor and OSSD to be an exploration activity. Rullani and Frederiksen [33] also adopted organizational learning theory to interpret exploitation as the contribution of code to existing projects and exploration as the creation of new projects using completely new code and knowledge. Therefore, we focused on the roles of existing developers and newcomers in OSSD projects in our own research.

3 | HYPOTHESES

In OSSD projects, external developers can provide new ideas and expert knowledge that enhance the project performance and solve problems encountered by the developers. When software is developed, the developers address certain issues, such as bugs and functional disabilities, that require professional knowledge. Ordinarily, the project cannot proceed to the next step if the organization members do not have the skill to solve the problems [34]. Thus, if the internal members lack the knowledge to move forward or if the contribution of new ideas from external developers is absent, the OSSD project may fail.

From the perspective of organizational learning theory, internal and external developers can positively impact the performance of the OSSD project. Research in the field of organizational learning theory emphasizes that exploration can provide many opportunities to produce innovative results that cannot be obtained using existing knowledge [11]. At the same time, although exploitation can result in a lack of creativity and new knowledge, it can also efficiently extend the existing capabilities of the company [35]. In this regard, both exploration and exploitation activities can enhance the OSSD project performance. Here, we present the following hypotheses:

Hypothesis 1-1 *Exploration has a positive impact on the project performance.*

Hypothesis 1-2 *Exploitation has a positive impact on the project performance.*

According to organizational learning theory, because exploration activities alone cannot ensure a successful outcome, the firm must also invest in exploitation. However, investing too much in one activity may result in a decrease in the firm's performance [36]. In previous research, the importance of using a combined approach, that is., the use of both exploration and exploitation, to reduce the uncertainty involved in relying on exploration for new ideas has been emphasized. This process is called ambidextrous learning [37].

Although it is difficult to find an optimal balance between exploration and exploitation [38], exploitation activities can reduce the uncertainty resulting from the exploration. However, most OSSD research focuses primarily upon the participation of new members and considers the success of OSSD projects to be dependent upon the participation of external developers [9]. In the same vein, few studies investigate the importance of internal developers. Moreover, research investigating the optimal ratio of external developers to the total number of developers or governance strategies to maximize their performance is lacking.

To validate the role of existing developers (eg, organization members), we studied how exploitation moderates the relationship between exploration and organizational performance. When developers begin to work together on certain OSSD, they often specify the purpose and expected functions of the software. Since the internal developers understand the sophisticated functions and final goals of the software, they can lead the collaboration in the right direction. Moreover, their expertise can facilitate working together to maintain and direct the OSSD. Therefore, the knowledge from internal developers who engage in exploitation will be helpful in moderating the influence of exploration.

Hypothesis 2 *Exploration has a stronger effect on the performance of projects whose focus is exploitation than it does on the performance of projects that depend heavily on the work of outside developers.*

According to organizational learning theory, the technology developed through exploration activities is maintained by developers internal to the organization if the technology is commercialized or needs continuous development [39]. In OSSD, the software developed through exploration and exploitation is often undergoing maintenance, and new versions of the software are being released. The developers' activities and roles change during the OSSD development process. For example, when the software has the desired functionality, fixing bugs becomes important.

Cooperation is easy for developers who fully understand that innovation involves the update and release of software with new features that are built upon basic functions. Before the OSS is released, the software may be dramatically changed by the developers, and/or new code may be adopted that changes the existing features [40]. However, after releasing the OSS, the primary goal is its maintenance [41]. Therefore, the number of dramatic changes in the software sharply decreases as the number of releases increases. The focus shifts to maintaining the software, which is done by internal developers who understand the software development process and the need for stability [41].

Thus, as the software development progresses, the impact of exploration on the organizational performance decreases. Mockus and others [42] stated that it is important to fix critical bugs so that the core project developers can manage the release of the software to prevent unexpected changes when it is released. We therefore hypothesize that the role of exploration decreases after releasing the software.

Hypothesis 3 *As the number of releases increases, the effect of exploration on the project performance decreases.*

4 | ANALYSIS

4.1 | Data

As a part of our research, we collected data from GitHub, a representative OSSD platform, using a web crawler that was developed using the Python programming language. We targeted the repositories (projects) owned by organizations in GitHub that have received more than five stars per week since the creation of the repository (the number of stars corresponds to the number of users who are interested in the repository). We selected repositories created between January and July of 2014 that have been operating for more than 300 days, which was long enough for an accurate assessment of the project results. To reinforce the meaning of cooperation, we selected projects with two or more developers [6]. In the end, we analyzed 17 691 repositories.

To study the repositories' performances, we considered the number of commits contributed to the project for each repository (*Commit*) to be a dependent variable. We assumed that commits were associated with the phases of the project during which developers were processing and enhancing the software. Therefore, we assumed that the number of commits represented the performance of the developers. Previous studies have used the number of commits contributed to the project or contributed by each developer to reflect the outcome of the project or individual, respectively [43–45].

The independent variables are *Exploitation* and *Exploration*, which represent the activities of organizational learning. *Exploitation* represents the percentage of members who participated in the specific project from among the total number of organization members, and *Exploration* represents the percentage of external developers from among the participating developers.

We used *Exploitation* and the number of software releases corresponding to a particular repository (*Release*) as moderating variables. The control variables are the total number of developers of the software associated with a repository (*Contributor*), the total number of members in an organization (*Member*), and the number of days the repository had been running (*RepoPeriod*). The variable definitions in this research model are summarized in Table 1 below.

4.2 | Analysis

The results of the descriptive statistics are shown in Table 2. The average commit number for the 17 691 repositories is 391 and the standard deviation is 2365.381. The histogram distribution confirms that the commit distribution does not follow a normal distribution. Following the estimation approach of linear regression analyses that has been used previously, we used a natural log transformation of Commit to obtain its skewness value [46,47].

The average number of developers working on a project from each repository is 7.3, and the average number of members in each organization is 85.5. The average percentage of the number of organization members who participated in a project (*Exploitation*) is 17.6%. The average percentage of external developers (*Exploration*) is 61.1%. This suggests that repositories owned by organizations have more external developers than they have internal developers. The average release count is 8.43.

Table 3 shows the correlation between the variables. The performance of a repository is positively correlated with the exploration (correlation = 0.2478; p-value < 0.01) and the exploitation (correlation = 0.2394; p-value < 0.01). Additionally, the exploration (correlation = 0.3278; p-value < 0.01) and the exploitation (correlation = 0.0693; p-value < 0.01) are positively correlated with the number of contributors in a repository, whereas the exploration (correlation = −0.0247; p-value < 0.01) and the exploitation (correlation = −0.3357; p-value < 0.01) are negatively correlated with the number of members in an organization.

We divided the repositories into two groups depending on whether the repository had released software or not. We then conducted a t test to compare their *Exploitation* and *Exploration* in Table 4. It appears that *Exploitation* does not vary significantly. However, *Exploration* is higher in the repositories with releases than it is in the repositories without releases. We see that additional input from the organization members does not increase after the release, whereas the voluntary participation of external developers does.

We performed a hierarchical regression using Stata 14 to test the hypotheses. The verification method of Baron and Kenny [48] was used to analyze the moderating effect. Parameters used in the mean centering were independent variables and interaction terms in the multicollinearity problem.

In Model 1, we used the operation period of the repository, the number of contributors, and the number of organization members as control variables. The results show that the more contributors who participated in a repository, the better the performance of the repository, whereas more members in an organization resulted in low performance for the organization's repository. Recall that we targeted repositories corresponding

TABLE 1 Description of variables

Variable		Definition
Dependent variable	*Commit*	Total number of commits in a repository
Independent and moderating variables	*Exploration*	Ratio of external developers to total contributors in a repository
	Exploitation	Ratio of organization members who contribute to a repository to total members in an organization
	Release	Number of software releases in a repository
Control variables	*Contributor*	Total number of developers in a repository
	Member	Total number of members in an organization
	RepoPeriod	Number of days since the repository was created

TABLE 2 Descriptive Statistics of Variables ($N = 17\,691$)

Variable	Min	Max	Mean	Standard deviation
Commit	2	173 145	391.263	2365.385
ln (Commit)	0.693	12.062	4.516	1.519
Exploitation	0	1	0.176	0.244
Exploration	0	1	0.611	0.288
Release	0	3043	8.436	43.281
RepoPeriod	337	708	531.124	103.735
Contributor	2	94	7.315	9.038
Member	2	976	85.469	159.701

TABLE 3 Correlation analysis

	1	2	3	4	5	6	7
1. Exploration	1.0000						
2. Exploitation	–0.1504**	1.0000					
3. Release	0.0633**	0.0249**	1.0000				
4. ln(Commit)	0.2478**	0.2394**	0.2249**	1.0000			
5. Contributor	0.3278**	0.0693**	0.1457**	0.5438**	1.0000		
6. Member	–0.0247**	–0.3357**	–0.0012	–0.0336**	0.0914**	1.0000	
7. RepoPeriod	0.0072	0.0017	0.0159*	0.0282**	0.0404**	–0.0041	1.0000

Note: $N = 17\,691$.

*$p < 0.05$;

**$p < 0.01$.

TABLE 4 Descriptive statistics and comparison of subsample means for repositories that have released software versus those that have not released software

Variable	Released software ($n = 9626$) M (SD)	Not released software ($n = 8065$) M (SD)	p-value
Exploitation	0.1733 (0.0027)	0.1789 (0.0025)	0.1295
Exploration	0.6247 (0.0028)	0.5952 (0.0033)	0.0000**

Note: Using the t test.

*$p < 0.05$;

**$p < 0.01$.

to projects with periods over 300 days; however, we found that this parameter does not affect the repository performance. The explanation power of Model 1 is 30.05%.

In Model 2, we added exploitation and exploration as independent variables. The higher the exploitation and exploration percentages, the higher the repository's commit value (H1-1 and H1-2 are supported). The explanation power of Model 2 is 35.13%, showing an increase of 5.08% p over Model 1.

In Model 3, we added exploitation and exploration as interaction terms to verify the moderating effect of exploitation. Exploration has a greater impact on the performance of projects with high exploitation values (H2 is supported). After analyzing the moderating effect of exploitation on the impact of exploration on a repository's performance, we found that the explanation power increases by 0.13% p.

In Model 4, we verified the moderating effect of the number of software releases by adding the release count and the interaction term of exploration and release count. We see that the release count moderates the impact of exploration on the number of a repository's commits. The exploration and release count interaction have a negative impact on the performance of a repository. This suggests that as the release count increases, the impact of exploration on the repository's performance decreases (H3 is supported). It appears that the explanation power increases by 2.82% p. The outputs from these hierarchical regression models are shown in Table 5.

5 | RESULTS AND DISCUSSION

We systemically examined the effects of openness in an OSSD project on the project's performance using organizational learning theory. In open collaboration, openness is known to

TABLE 5 Output of hierarchical regression model

	Model 1		Model 2		Model 3		Model 4	
Variable	**Coefficient**	**SE**	**Coefficient**	**SE**	**Coefficient**	**SE**	**Coefficient**	**SE**
RepoPeriod	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000
Contributor	0.0930**	0.0011	0.0825**	0.0011	0.0813**	0.0011	0.0783**	0.0011
Member	–0.0008**	0.0001	0.0000	0.0000	0.0000	0.0000	–0.0000	0.0000
Exploitation			0.6353**	0.0347	0.6582**	0.0349	0.5771**	0.0344
Exploration			1.3901**	0.0412	1.4342**	0.0418	1.4067**	0.0409
Exploitation × Exploration					0.7840**	0.1298	0.7672**	0.1270
Release							0.0091**	0.0003
Release × Exploration							–0.0174**	0.0011
R^2	0.3050		0.3513		0.3526		0.3809	
Adjusted R^2	0.3048		0.3511		0.3524		0.3806	
ΔF			631.559**		36.484**		404.237**	
N	17 691		17 691		17 691		17 691	

*$p < 0.05$;

**$p < 0.01$ are the dependent variables for ln(*Commit*).

be an important component of innovation because outside developers collaborate with developers inside the organization. It is meaningful to examine how collaboration and the stage of software development affect a project's performance.

By building on previous research, we found that the exploration and exploitation ware both positively correlated with the performance of an open collaboration. Moreover, we determined that the impact of exploration increases with an increase in exploitation, that is, ambidextrous research has a positive impact on the project performance of an open collaboration over the Internet. In Table 6, the results of the hypothesis testing are summarized. We see that in open collaboration, external members of an organization can effectively collaborate with internal members. However, it is not desirable to emphasize openness simply to maximize the performance of an open collaboration. Governance, which emphasizes a balance between exploration and exploitation, is needed in open collaboration.

In addition, since the roles of the internal and external developers in OSSD projects dramatically changes depending on the number of releases, we investigated how the effect of exploration on project performance changes depending on the software-development stage. Results show that the impact of exploration on the project performance decreases after releasing the OSS. Thus, we also assert that it is most effective to absorb external knowledge in the early stages of innovation rather than at a later stage.

6 | CONCLUSION

We investigated the factors affecting the performance of open collaborations involving outside developers from the perspective of organizational learning theory. We conclude that participation from outside developers enhances project performance, which is measured by the number of commits, provided outside developers collaborate with the internal developers. Moreover, we clarified the roles of exploration and exploitation by considering the mutuality of software. This paper contributes to the field of knowledge creation by empirically analyzing the effects of exploration on innovation.

We examined empirical data from many projects. In most previous research, one or two large exploratory projects were examined. In this study, we gathered data from the GitHub coding platform. Over 17 691 repositories were included in our data set.

TABLE 6 Results of hypotheses test

No.	Hypotheses	Results
1-1	Exploration will have a positive impact on the project performance	Supported
1-2	Exploitation will have a positive impact on the project performance	Supported
2	Exploration has a stronger effect on the performance of projects with high exploitation than on projects with low exploitation	Supported
3	As the number of releases increases, the effect of exploration on the project performance will decrease	Supported

This study has empirical applications. The results of this study contribute to the development of government policies regarding the most effective ratio of internal to external developers on a project. The results show that an organization can control the internal developer participation based on the extent to which the external developers participate. In addition, as the number of releases increases, our research suggests that as the project evolves, the ratio of internal to external developers increases.

This study has some limitations. First, by measuring the performance of a project based on its commits, we cannot consider the quality of the OSS. We can, however, quantify how much the OSS has changed through using the number of commits as an indicator of the amount of work that has been done on the software and the overall contributions from the developers. In the future, we wish to consider the quality of the software and the impacts of organizational learning activities on project performance as explanatory variables. Moreover, we wish to conduct an in-depth analysis of the communication patterns between the internal and external developers to determine their exploration and exploitation activities. Finally, we wish to refine our use of commits by considering their purposes and functions, that is, by identifying commits that fix bugs, those that change features of the software, and so on.

ORCID


Hyunmi Baek https://orcid.org/0000-0001-5995-2565

REFERENCES

1. S. L. Brown and K. M. Eisenhardt, *The art of continuous change: Linking complexity theory and time-paced evolution in relentlessly shifting organizations*, Administrative Sci. Quarterly **42** (1997), no. 1, 1–34.
2. F. T. Rothaermel and D. L. Deeds, *Exploration and exploitation alliances in biotechnology: A system of new product development*, Strategic Manag. J. **25** (2004), no. 3, 201–221.
3. S. S. Levine and M. J. Prietula, *Open collaboration for innovation: Principles and performance*, Organization Sci. **25** (2013), no. 5, 1414–1433.
4. S. Chengalur-Smith and A. Sidorova, *Survival of open-source projects: A population ecology perspective*, in Proc. Int. Conf. Inf. Syst., Seattle, WA, USA, 2003, 782–787.
5. A. Lima, L. Rossi, and M. Musolesi, *Coding together at scale: GitHub as a collaborative social network*, in Proc. AAAI Int. Conf. Weblogs Social Media, Copenhagen, Denmark, July 2014, pp. 31–40.
6. S. R. Lee, H. M. Baek and J. J. Jahng, *Governance strategies for open collaboration: Focusing on resource allocation in open source software development organizations*, Int. J. Inf. Manag. **37** (2017), no. 5, 431–437.
7. M. Markus. *The governance of free/open source software project: Monolithic, multidimensional, or configurational?* J. Manag. Governance **11** (2007), no. 2, 151–163.
8. W. Scacchi and C. Jensen, *Governance in open source software development projects: Towards a model for network-centric edge organizations*, DTIC Document, 2008.
9. J. Hahn, J. Y. Moon, and C. Zhang, *Emergence of new project teams from open source software developer networks: Impact of prior collaboration ties*, Inf. Syst. R. **19** (2008), no. 3, 369–391.
10. Z. L. He and P. K. Wong, *Exploration vs. exploitation: An empirical test of the ambidexterity hypothesis*, Org. Sci. **15** (2004), no. 4, 481–494.
11. J. G. March, *Exploration and exploitation in organizational learning*, Org. Sci. **2** (1991), no. 1, 71–87.
12. L. Dabbish et al., *Social coding in GitHub: Transparency and collaboration in an open software repository*, in Proc. ACM Conf. Comput. Supported Cooperative Work, Seattle, WA, USA, Feb. 2012, pp. 1277–1286.
13. B. Perens, The open source definition. Open sources: Voices from the open source revolution, O'Reilly & Associates, Inc., Sebastopol, CA, USA, 1, 1999, pp. 171–188.
14. G. Gousios and D. Spinellis, *GHTorrent: GitHub's data from a firehose*, in Proc. IEEE Working Conf. Mining Softw. Repositories, Zurich, Switzerland, June 2012, pp. 12–21.
15. A. Kieser, N. Beck, R. Tainio, *Rules and organizational learning: The behavioral theory approach*, Handbook of Organizational Learning and Knowledge (M. Dierkes, A. Antal, J. Child and I. Nonaka, eds.), Oxford University Press, New York, 2001.
16. B. Levitt and J. G. March, *Organizational learning*, Annu. Rev. Soc. **14** (1988), no. 1, 319–338.
17. N. Argyres, *Capabilities, technological diversification and sivisionalization*, Strategic Manag. J. **17** (1996), no. 5, 395–410.
18. G. Ahuja and C. M. Lampert, *Entrepreneurship in the large corporation: A longitudinal study of how established firms create breakthrough inventions*, Strategic Manag. J. **22** (2001), 521–543.
19. M. J. Benner and M. Tushman, *Process management and technological innovation: A longitudinal study of the photography and paint industries*, Administrative Sci. Quarterly **47** (2002), no. 4, 676–707.
20. G. Dowell and A. Swaminathan, *Entry timing, exploration, and firm survival in the early US bicycle industry*, Strategic Manag. J. **27**, (2006), no. 12, 1159–1182.
21. R. Katila and G. Ahuja, *Something old, something new: A longitudinal study of search behavior and new product introduction*, Academy Manag. J. **45** (2002), no. 6, 1183–1194.
22. A. Nerkar, *Old is gold? The value of temporal exploration in the creation of new knowledge*, Manag. Sci. **49** (2003), no. 2, 211–229.
23. L. Rosenkopf and A. Nerkar, *Beyond local search: Boundary spanning, exploration, and impact in the optical disk industry*, Strategic Manag. J. **22** (2001), no. 4, 287–306.
24. R. Belderbos *et al.*, *Technological activities and their impact on the financial performance of the firm: Exploitation and exploration within and between firms*, J. Product Innovation Manag. **27** (2010), no. 6, 869–882.
25. S. Jayanthi and K. K. Sinha, *Innovation implementation in high technology manufacturing: A chaos-theoretic empirical analysis*, J. Operat. Manag. **16** (1998), no. 4, 471–494.
26. A. Nerkar and P. W. Roberts, *Technological and Product market experience and the success of new product introductions in the pharmaceutical industry*, Strategic Manag. J. **25** (2004), no. 8–9, 779–799.
27. B. Van Looy, T. Martens, and K. Debackere, *Organizing for continuous innovation: On the sustainability of ambidextrous organizations*, Creativity Innovation Manag. **14** (2005), no. 3, 208–221.
28. A. Phene, K. F. Lindquist, and L. Marsh, *Breakthrough innovations in the US biotechnology industry: The effects of technological space and geographic origin*, Strategic Manag. J. **27** (2006), no. 4, 369–388.
29. J. S. Sidhu, H. R. Commandeur, and H. W. Volberda, *The multifaceted nature of exploration and exploitation: Value of supply, demand, and spatial search for innovation*, Org. Sci. **18** (2007), no. 1, 20–38.

30. F. Perretti and G. Negro, *Mixing genres and matching people: A study in innovation and team composition in Hollywood*, J. Organizational Behavior. **28** (2007), no. 5, 563–586.
31. J. A. Chatman, *Improving interactional organizational research: A model of person-organization fit*, Academy Manag. Rev. **14** (1989), no. 3, 333–349.
32. M. Osterloh and S. Rota, *Open source software development—Just another case of collective invention?* Res Policy **36** (2007), no. 2, 157–171.
33. F. Rullani and L. Frederiksen, *Individual interaction and innovation capabilities: Exploration and exploitation in open source software communities*, in Proc. DRUID Summer Conf. Opening Innovation: Strategy, Organization Technol., London, UK, June 2010, pp. 1–52.
34. J. Howison and K. Crowston, *Collaboration through open superposition: A theory of the open source way*, MIS Quarterly **38** (2014), no. 1, 29–50.
35. G. B. Voss and Z. G. Voss, *Strategic ambidexterity in small and medium-sized enterprises: Implementing exploration and exploitation in product and market domains*, Org. Sci. **24** (2013), no. 5, 1459–1477.
36. V. Gilsing and B. Nooteboom, *Exploration and exploitation in innovation systems: the case of pharmaceutical biotechnology*, Res. Policy **35** (2006), no. 1, 1–23.
37. M. L. Tushman and C. A. O'Reilly III, *Ambidextrous organizations: Managing evolutionary and revolutionary change*, California Manag. Rev. **38** (1996), no. 4, 8–29.
38. D. A. Levinthal and J. G. March, *The myopia of learning*, Strategic Manag. J. **14** (1993), no. S2, 95–112.
39. A. E. Eiben and C. A. Schippers, *On evolutionary exploration and exploitation*, Fundamenta Informaticae **35** (1998), 35–50.
40. O. Baysal and A. J. Malton, *Correlating social interactions to release history during software evolution*, in Proc. Int. Workshop Mining Softw. Repositories, Mineapolis, MN, USA, May 2007, pp. 1–8.
41. Q. Tu, *Evolution in open source software: A case study*, in Proc. IEEE Int. Conf. Soft. Maintenance, San Jose, CA, USA, Oct. 2000, pp. 131–142.
42. A. Mockus, R. T. Fielding, and J. Herbsleb, *A case study of open source software development: The Apache server*, in Proc. Int. Conf. Soft. Eng., Limerick, Ireland, June 2000, pp. 263–272.
43. G. Ahuja and C. Morris Lampert, *Entrepreneurship in the large corporation: A longitudinal study of how established firms create breakthrough inventions*, Strategic Manag. J. **22** (2001), no. 6–7, 521–543.
44. K. Crowston, H. Annabi, and J. Howison, *Defining open source software project success*, in Proc. Intern. Conf. Inf. Syst., Seattle, WA, USA, Dec. 2003, pp. 1–14.
45. H. Baek and S. Oh, *Identifying the network characteristics of contributors that affect performance in open collaboration: Focusing on the GitHub open source*, J. Soc. e-Business Studies **20** (2015), no. 1, 23–43.
46. V. Dhar and E. A. Chang, *Does chatter matter? The impact of user-generated content on music sales*, J. Interactive Marketing **51** (2009), no. 4, 300–307.
47. W. Duan, B. Gu, and A. Whinston, *The dynamic of online word-of-mouth and product sales: And empirical investigation of the movie industry*, J. Retailing **84** (2008), no. 2, 233–242.
48. R. M. Baron and D. A. Kenny, *The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations*, J. Personality Soc. Psychol. **51** (1986), no. 6, 1173–1182.

AUTHOR BIOGRAPHIES

Saerom Lee received her BS in international trade from Pusan National University in 2010 and her PhD in management information systems from Seoul National University, Seoul, Republic of Korea in 2016. Since 2018, she has been an assistant professor in the School of Business Administration, Kyungpook National University, Daegu, Republic of Korea. Her main research interests are open collaboration in open source software development and online loafing behaviors with a focus on sexual harassment.

Hyunmi Baek received her BS in chemical engineering from the Pohang University of Science and Technology, Pohang, Republic of Korea in 2002 and her MS in IT management from the Information and Communications University, Daejeon, Republic of Korea in 2004. From 2003 to 2013, she worked for Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea. She received her PhD in management information systems from Seoul National University, Seoul, Republic of Korea in 2013. From 2013 to 2018, she worked for Hanyang University. Since 2018, she has been an assistant professor in the School of Media and Communication, Korea University, Seoul, Republic of Korea. Her primary research interests are electronic word-of-mouth in social media, information adoption, and open collaboration.

Sehwan Oh earned his MS in information technologies (MSIT) from the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA in 2007 and his BA and PhD in economics and business administration from the College of Social Sciences and the College of Business Administration, Seoul National University, Seoul, Republic of Korea in 2001 and 2015, respectively. From 2003 to 2015, he worked for the Korea International Trade Association, Seoul, Republic of Korea. Since 2015, he has been an assistant professor in the School of Business Administration, Kyungpook National University, Daegu, Republic of Korea. His current research interests include sharing economies, electronic word-of-mouth, and e-commerce. He has published on these topics in journals, such as Internet Research, the International Journal of Mobile Communications, the Journal of Electronic Commerce Research, ETRI Journal, and the Asia Pacific Journal of Information Systems.