Once Bitten, Twice Shy: The Influence of Gatekeepers on Contributors' Continued Participation in Open Source Software Projects

ABSTRACT

The success of firms' participation in open source software (OSS) development depends heavily upon their ability to maintain a constant flow of voluntary contributions. The emerging literature on contributors' continued participation suggests that contributors' willingness to stay in OSS projects is shaped by their experience of prior participation. We extend this literature by examining how gatekeepers' interactions with contributors influence their continued participation in firm-hosted OSS projects. Predictions are formalized in an analytical model that analyzes how contributors' belief updating process is affected by gatekeepers' code acceptance and knowledge sharing activities. By utilizing data on 9,271 newcomers' participation in Google-owned OSS projects on GitHub, we find that gatekeepers' knowledge sharing and code acceptance in initial interactions have positive effects on newcomers' continued participation in OSS projects. In addition, these effects persist among newcomers who work for the OSS firms and are magnified if gatekeepers have high expertise.

Keywords: Open source software (OSS), knowledge sharing, gatekeeping, innovation, online community.

INTRODUCTION

Firms increasingly participate in open source software (OSS) development (Ho & Rai, 2017; Jeppesen & Frederiksen, 2006; Spaeth, von Krogh, & He, 2015). For instance, Microsoft, which used to view OSS as a threat to its business model, has become one of the most active contributors on GitHub, the world's largest code hosting platform (Asay, 2018). By leveraging voluntary contributions from OSS communities, firms can build innovative capabilities, promote product adoption and reduce development cost (Dahlander & Magnusson, 2005; Daniel, Maruping, Cataldo, & Herbsleb, 2018; West, 2003). As a result, the success of firms' participation in OSS development significantly depends upon their ability to maintain a constant flow of voluntary contributions (Zhang, Hahn, & De, 2013), which requires firms to retain existing contributors in their OSS projects instead of merely seeking one-off contributions.

Despite the importance of sustaining OSS contributors' participation, only a small number of studies have focused on factors influencing contributors' continued participation in OSS projects (Dejean & Jullien, 2015; Fang & Neufeld, 2009; von Krogh, Haefliger, Spaeth, & Wallin, 2012). This stream of studies suggests that factors affecting continued participation may differ from those that drive initial participation, since some of the benefits from participating in OSS projects are available only after the contribution has been made (Shah, 2006; von Hippel & von Krogh, 2003). Specifically, contributors' willingness to stay in OSS projects is shaped by their experience of prior participation, such as social interaction with the community (Fang & Neufeld, 2009), accumulated reputation (Hann, Roberts, & Slaughter, 2013), or perceived project quality (Ho &Rai, 2017). However, prior research generally only looks at the influence of community or project level factors on contributors' participation, while paying limited attention to how firm employees may act as gatekeepers to regulate and influence external contributions

(Dahlander & O'Mahony, 2011; Massa & O'Mahony, 2021). Filling this gap is important because gatekeepers often actively interact with contributors in the OSS development process and therefore their behaviors may exert an influence on contributors' motivations to continue their participation.

We aim to address this gap by focusing on how gatekeepers' interactions with contributors influence their continued participation in firm-hosted OSS projects. To be specific, we focus on two gatekeeper behaviors: (1) knowledge sharing, and (2) code acceptance. Gatekeepers' knowledge-sharing behaviors in their review process can benefit contributors substantially by helping the latter extend programming skills and improve the quality of their codes (Riedl & Seidel, 2018; Singh, Tan, & Youn, 2011; von Krogh, Haefliger, Spaeth, & Wallin, 2012). Similarly, contributors also like to see their work implemented to exert influence on OSS projects, accumulate reputation, and receive potential opportunities from the OSS firm (Acar, 2019; Fang et al., 2009; von Krogh, Spaeth, & Lakhani, 2003; Zaggl, 2017). Nevertheless, contributors do not always obtain these benefits in their participation because of the uncertainty regarding gatekeepers' review patterns, which makes it difficult for contributors to form participation decisions. Building upon the belief updating framework from behavioral economics (Augenblick & Rabin, 2021; Benjamin, 2019), we argue that contributors' decision about whether to continue their participation is based on their future beliefs about gatekeepers' knowledge sharing and code acceptance activities, and that such beliefs depend on their prior interactions with gatekeepers. Through direct interactions with gatekeepers, contributors are able to gain private information regarding gatekeepers' behavioral patterns or preferences, and use this information to form more accurate expectations. In particular, newcomers (contributors who contribute to the focal OSS projects for the first time) rely heavily on initial interactions to form future beliefs due to a lack of prior experience. As a result, we expect that both knowledge sharing and code acceptance in initial interactions will have positive effects on newcomers' continued participation in OSS projects. Moreover, we expect that these positive effects will be further shaped by gatekeepers' expertise, and newcomers' organizational affiliation, as gatekeepers with high expertise are more likely to share performance-enhancing knowledge and newcomers who work for the OSS firm may receive limited benefits from code implementation.

To formalize the above arguments, we develop a simple analytical model to unfold contributors' belief updating process and analyze the influence of associated contingency factors. A set of hypotheses are derived from this model and are tested by examining newcomers' participation in Google-owned OSS projects on GitHub. Google, as one of the most influential companies engaging in OSS development, hosts more than two thousand public repositories (OSS projects) on GitHub, granting free access to all users and encouraging contributors to collaborate on its repositories. Google attaches great importance to the interactions between gatekeepers and contributors, and has published a set of engineering practices that guide the behaviors of gatekeepers (see Kammer, Hodges, & Murillo, 2019), helping to open up the black box of the code review process (Sadowski, Söderberg, Church, Sipko, & Bacchelli, 2018). Our data contain detailed information on newcomers' contribution activities (submission of pull requests) in Google-owned repositories, as well as gatekeepers' code implementation decisions (merge) and comments on newcomers' work. Using these data, we find that newcomers who receive knowledge from gatekeepers or whose code is accepted are more likely to continue their participation in the focal OSS projects. These positive effects persist even among newcomers who work for the OSS firm. In addition, the positive influence of gatekeepers' knowledge sharing is stronger if gatekeepers have high expertise. Further, to explore the spillover effects of gatekeepers' interactions on newcomers' participation in other firm-hosted OSS projects, we also examine whether knowledge sharing and code acceptance motivate newcomers to submit a second pull request to any projects owned by the OSS firm, and find strong evidence of such influence.

Our study makes several contributions to the literature. First, we contribute to research on contributors' participation in firm-hosted OSS projects by focusing sharply on the critical role of gatekeepers, the people representing the OSS firm, which was neglected in prior studies. Second, this study provides a belief-updating perspective on how prior interactions will affect future participation. Third, we also contribute to research and practice on OSS project governance by highlighting the importance for firms to regulate and guide gatekeepers' behaviors in their review process.

THEORY AND HYPOTHESES

Continued participation, gatekeeper, and belief updating

Contributors' continued participation plays an important role in the success of firm-hosted OSS projects. Given contributors' weak dependence on OSS projects and the absence of direct pecuniary rewards, it is important for OSS firms to provide sufficient incentives to retain existing contributors (Alexy, West, Klapper, & Reitzig, 2017; Chen, Tong, Tang, & Han, 2022). Continued participation differs from initial participation in that contributors will adjust their perceived benefits from participation according to their prior experiences in the OSS development process. Therefore, contributors' willingness to stay in OSS projects depends significantly on factors derived after the initial participation (Karahanna, Straub, & Chervany, 1999; Zhang et al., 2013). Prior studies on OSS projects have examined what mechanisms influence contributors' continued participation by creating links between various motives and contributions. For example, Shah (2006) provided empirical evidence that developers who derived enjoyment from open source contributions were more likely to engage in sustained participation. Spaeth, von Krogh, and He (2015) focused on specific firm attributes and found that firms' credibility and openness enhanced volunteers' intrinsic motivation. In addition, several studies have shown that project quality played a crucial role in promoting contributors' continued participation (Baldwin & Clark, 2006; Ho et al., 2017).

However, research has given limited attention to how gatekeepers' interactions with contributors may affect the latter's continued participation. OSS firms do not unconditionally accept contributions from all voluntary contributors, and rely on firm members to manage OSS projects and filter external information. Gatekeepers of an OSS firm refer to individuals who are responsible for evaluating and monitoring voluntary contributions to firm-hosted OSS projects to ensure the quality of these contributions (Allen & Cohen, 1969; Massa et al., 2021). They are crucial to ensuring the overall code health of firm-hosted OSS projects. Specifically, gatekeepers often actively interact with contributors in the code review process, guide contributors on how to meet the required standards, and decide on whether to accept contributors' work. Such direct interactions provide an important conduit for contributors to garner private benefits such as learning and reputation accumulation (von Hippel & von Krogh, 2003). In addition, through communication with gatekeepers, contributors will form impressions about gatekeepers' attitudes or preferences toward their contributions, which may also exert an impact on contributors' willingness to continue their participation.

The remaining question then is how and to what extent will gatekeepers' activities affect contributors' continued participation. Given our context, it would be helpful to apply a belief updating framework to illustrate how contributors' beliefs about the benefit of OSS participation will be affected by gatekeepers' activities and how such evolution will influence their future participation. Belief updating is an important field of research in behavioral economics that examines how individuals adjust their expectations in the face of new information (Charness & Levin, 2005; Zimmermann, 2020; Scharfstein & Stein, 1990). The belief updating process generally involves two stages (Ambuehl & Li, 2018). First, individuals update beliefs about the probability of certain events based on new evidence they observe (e.g., by following the Bayes' rule). They then make decisions that yield the highest expected utility conditional on their beliefs (Camerer & Loewenstein, 2011). Such updating process is subjective and depends on individual characteristics. Following this logic, we expect that open source contributors will utilize their experiences in current participation to update their beliefs about the probability of gatekeepers' interactions in the subsequent participation, and will choose to continue to participate only if the expected benefits from gatekeepers' interactions are sufficiently high.

A simple belief updating model

To formalize the above arguments, we develop a simple analytical model to unfold contributors' belief updating process and analyze the influence of associated contingency factors. We also generate several predictions that underscore the important role of gatekeepers' behaviors in affecting contributors' continued participation.

Consider an open source contributor who has contributed to an OSS project and is deciding whether to continue his participation. The contributor's benefits from participating in the OSS project depend on whether he receives knowledge from the gatekeeper and whether his work is accepted by the gatekeeper. For simplicity, we assume constant costs required to complete his work, and the following equation illustrates the individual utility function:

$$U_i = \alpha M_i + \beta G_i - C \tag{1}$$

where M_i measures the amount of knowledge received by the contributor, G_i captures the code acceptance decision of the gatekeeper ($G_i = 1$ if the code is implemented, and $G_i = 0$ if the code is rejected), C represents the constant cost, α and β measure the unit-level utility gained from knowledge sharing and code acceptance.

Next, consider the belief updating process when uncertainty exists about the probability of the gatekeeper's knowledge sharing and code acceptance behaviors in the subsequent participation. Following the work of Lochner (2007), we assume that the contributor's belief depends on his prior experiences of interacting with the gatekeeper, and the belief updating rules for knowledge sharing and code acceptance are:

$$E(M_i|H_i^t) = g(E(M_i|H_i^{t-1}), M_{it}) \quad (2)$$

$$P(G_i = 1|H_i^t) = f(E(M_i|H_i^{t-1}), M_{it}, G_{it}) \quad (3)$$

where H_i^t represents the information available at date t, M_{it} and G_{it} measure the gatekeeper's knowledge sharing and code acceptance behaviors at date t. Note that the expected probability of code acceptance in the continued participation also depends on the gatekeeper's prior knowledge-sharing activities. This is because by receiving knowledge from the gatekeeper, the contributor is more likely to improve code quality in the continued contribution and therefore his code submission is more likely to be accepted. To specify the belief updating process, we rewrite equations (2) and (3) as:

$$E(M_i|H_i^t) = \frac{a}{a+b} M_{i,t} + \frac{b}{a+b} m_{i,t-1} \quad (4)$$

$$P(G_i = 1|H_i^t) = \frac{c}{c+d} G_{i,t} + \frac{d}{c+d}\ g_{i,t-1} + em_{i,t} \quad (5)$$

where $m_{i,t-1}$ and $g_{i,t-1}$ represent the contributor's prior beliefs about the probability of knowledge sharing and code acceptance at date t-1. Combine equations (4) and (5), and we can derive the expression for the contributor's expected utility in the continued participation as well as the marginal influence of prior knowledge sharing and code acceptance behaviors:

$$E(U_{it}|H_i^t) = \left(\alpha \frac{a}{a+b} + \beta e\right) M_{i,t} + \alpha \left(\frac{b}{a+b}\right) M_{i,t-1} + \left(\beta \frac{c}{c+d}\right) G_{i,t} + \beta \left(\frac{d}{c+d}\right) G_{i,t-1} - C \quad (6)$$

$$\frac{dE(U_{it}|H_i^t)}{dM_{i,t}} = \alpha \frac{a}{a+b} + \beta e \quad (7)$$

$$\frac{dE(U_{it}|H_i^t)}{dG_{i,t}} = \beta \frac{c}{c+d} \quad (8)$$

Equation (7) and (8) have several implications for how gatekeepers influence contributors' continued participation, which we formalize as four propositions below:

Proposition 1: *Contributors who receive knowledge from gatekeepers are likely to continue their participation if* α *and* a *are large.*

Proposition 2: *Contributors whose work is accepted by gatekeepers are likely to continue their participation if* β *and* c *are large.*

Proposition 3: *The influence of gatekeepers' knowledge sharing on contributors' continued participation will be moderated if the received knowledge has only a minor influence in increasing the probability of code acceptance (when* e *is small).*

Proposition 4: *If contributors care less about whether their work is accepted (when* β *is small), then the influence of gatekeepers' knowledge sharing on contributors' continued participation will also be moderated.*

Proposition 1 and Proposition 2 show that gatekeepers' behaviors will positively influence contributors' continued participation if contributors can garner benefits from these behaviors and if contributors rely heavily on their latest experiences to predict gatekeepers' behaviors in the continued participation. Proposition 3 suggests that sharing knowledge that cannot significantly help contributors to achieve better performance in the OSS projects are less likely to motivate continued participation, since contributors may not be able to obtain the benefits of code acceptance. Proposition 4 shows that if contributors are less sensitive to whether their work is accepted, then their needs for receiving performance-enhancing knowledge will also decline, since their expected utility will not increase significantly even if their code is accepted.

Hypotheses

As mentioned above, prior studies have provided evidence that open source contributors enjoy the benefits of knowledge sharing and code acceptance (von Krogh et al., 2012; von Hippel et al., 2003). In addition, we expect newcomers to depend heavily on initial interactions to predict gatekeepers' behavior patterns in the continued participation. Unlike established contributors who have made multiple contributions in the OSS projects and have obtained sufficient private information regarding gatekeepers' preferences, newcomers generally have limited information to form accurate expectations before the first contribution (Morrison, 1993), and therefore rely more on initial interactions to gather information and update their beliefs. Based on these arguments and Propositions 1 and 2, we have the following hypotheses:

Hypothesis 1: *Newcomers who receive knowledge from gatekeepers are more likely to continue their participation.*

Hypothesis 2: *Newcomers whose work is accepted by gatekeepers are more likely to continue their participation.*

Proposition 3 suggests that the influence of knowledge sharing on newcomers' continued participation will be moderated if the knowledge cannot help improve newcomers' subsequent performance in the OSS projects. Following this, we expect that gatekeepers' expertise will influence the effect of their knowledge sharing on newcomers' continued participation, since experts are more likely to share high-quality knowledge that helps improve newcomers' subsequent performance (Kim & Viswanathan, 2019). In addition, equation (8) shows that providing performance-enhancing knowledge will not affect how prior code acceptance helps predict future code acceptance patterns, so we have:

Hypothesis 3a: *The influence of gatekeepers' knowledge sharing on newcomers' continued participation is stronger if gatekeepers have high expertise.*

Hypothesis 3b: *The influence of gatekeepers' code acceptance on newcomers' continued participation does not depend on gatekeepers' expertise.*

In a similar vein, we expect newcomers' affiliations with the OSS firm will affect their perceived benefits from code acceptance, since everything else constant, newcomers who already work for the firm have more limited motivations to signal their quality or are less likely to receive job opportunities from the firm. Based on Proposition 2 and Proposition 4, we have the following hypotheses:

Hypothesis 4a: *If newcomers work for the OSS firm, then the influence of gatekeepers' knowledge sharing on newcomers' continued participation will be moderated.*

Hypothesis 4b: *If newcomers work for the OSS firm, then the influence of gatekeepers' code acceptance on newcomers' continued participation will be moderated.*

DATA AND METHODS

Context

We focus on Google-owned OSS projects on GitHub. GitHub, as the world's largest OSS development platform, hosts more than 28 million public repositories (OSS projects) and 3 million organization users. An organization user on GitHub usually owns several public repositories, with each repository maintained by a group of organization members (maintainers).

Contributors can make contributions to organizations' repositories by copying the focal repositories into their accounts (fork), making changes to the code, and then submitting these changes to the repository (pull request). One or several repository maintainers will then be assigned to review contributors' pull requests, leave comments to contributors, and decide whether to accept their work (by merging the pull request or rejecting it). If the code is accepted, a "commit" will be created and added to the repository. In this study, we focus specifically on repositories owned by Google, because 1) Google is one of the most active firm users on GitHub and thus provides a sufficiently large sample for analyzing interactions between gatekeepers and contributors; and 2) Google has published engineering practices that guide the behaviors of gatekeepers (Kammer et al., 2019), which can help us to open the black box of the code review process and generate appropriate variables.

Sample construction

We utilized GitHub's Pull Request API to obtain information on contributors' participation in Google's public repositories. First, to construct the base sample, we collected data on all pull requests submitted to Google's public repositories. Second, since we are only interested in newcomers who rely on initial participation to form their beliefs, we need to exclude repository maintainers who contribute to their repositories, because repository maintainers can easily gather information regarding their team members. To do so, we dropped contributors who initially sent pull requests to repositories in which they have previously reviewed at least one pull request. In addition, since repository maintainers can also directly make code changes in their repositories without initiating a pull request, we dropped contributors who initially sent pull requests to repositories in which they have previously made at least one commit. Third, some newcomers' initial pull requests were reviewed by more than one repository maintainer, and we excluded them in our analysis because it is difficult to identify the influence of each repository maintainer. Forth, we noticed that several newcomers submitted their second pull requests to the repositories immediately after their initial pull requests (within 24 hours), and we excluded these newcomers because their continued participation may result from contributors' tendency to split a large pull request into several small ones. Consequently, the sample consisted of 10,639 newcomers and 1,266 gatekeepers.

Dependent variable

Continued participation. The dependent variable measured whether a newcomer submitted a second pull request to the focal repository. This was coded as a binary variable that took the value of 1 if the newcomer submitted a second pull request following his initial contribution.

Explanatory variables

Knowledge sharing. Gatekeepers can communicate and share knowledge with newcomers by leaving comments in the review process. Following prior work on knowledge sharing, we define knowledge-sharing comments as comments that provide information, opinions, or suggestions regarding newcomers' performance (Bartol & Srivastava, 2002). To identify such comments, we utilized a text analysis approach called dialogue act (DA) classification (Kim, Cavedon, & Baldwin, 2010), which classifies an utterance with respect to the function it serves in a dialogue. Specifically, for each sentence in gatekeepers' comments, we used a deep learning DA classifier to assign a label to the sentence to describe the function it served (see Table 1 for examples of labeled utterances). After evaluating the classification outcomes, we kept sentences labeled "Statement-opinion" since these sentences were likely to provide information or suggestions to newcomers. In addition, we created a word list containing suggestive phrases that may be related to knowledge sharing, and performed a keyword matching to target other sentences. To achieve satisfactory accuracy, we followed an iterative process: each time we randomly selected 500 knowledge-sharing sentences and hired two coders to independently label these sentences. A sentence was deemed correctly classified if both coders labeled it as "knowledge sharing", and the keyword matching method was repeatedly adjusted until the accuracy rate was above 70%. Knowledge sharing was thus computed as the total number of these sentences.

Code acceptance. Code acceptance was coded as a binary variable that took the value of 1 if the pull request was accepted.

Newcomer affiliation. Newcomer affiliation was coded as a binary variable that took the value of 1 if the newcomer was a Google employee.

Gatekeeper expertise. We computed a gatekeeper's expertise by calculating the total number of forks received by the gatekeeper.

Control variables

We included several control variables that might affect newcomers' motivations to continue their participation. Prior research has suggested that contributors' interactions with the community will affect their willingness to continue their participation (Fang et al., 2009). As a result, we controlled for newcomers' activities within the community, including the cumulative number of repos owned by newcomers in each month (*Newcomer Repo*) and the cumulative number of stars given by newcomers in each month (*Newcomer Star Given*). We also controlled for the number of months between newcomers' joined date and their initial participation in Google-owned repositories (*Newcomer Age*). At the pull request level, we incorporated several control variables for newcomers' effort level, since newcomers' investment in the initial pull request may affect their willingness toward continued participation. Specifically, we included the number of words in each pull request's description (*Description Length*), the number of commits made in pull requests (*Pull Request Commit*), as well as the average row number of code changes in each pull request' commits (*Code Change*). We also accounted for the number of days required to complete a pull request (*Review Time*) and the number of comments given by other contributors (*External Comment*). In addition, at the gatekeeper level, we measured gatekeepers' review history (*Gatekeeper Experience*) by computing the number of pull requests reviewed by a gatekeeper before newcomers' initial participation. We also created a dummy variable to control for whether gatekeepers have directly edited newcomers' pull requests (*Direct Edit*), since newcomers who receive direct edits from gatekeepers may be motivated to continue their participation. Lastly, we controlled for repository level variables by including the monthly cumulative number of forks received by the repository (*Repo Fork Received*), the monthly cumulative number of pull requests submitted to the repository (*Repo Pull Request Received*), as well as the number of months between the repository's creation date and newcomers' initial participation (*Repo Age*).

Estimation

We employed the Cox proportional hazard model with time-varying covariates to estimate newcomers' continued participation. However, our analysis may be subject to endogeneity issues, since 1) unobserved quality regarding the pull request may influence both gatekeepers' willingness to share knowledge as well as gatekeepers' code acceptance decision; and 2) there may exist a correlation between gatekeepers' knowledge sharing and code acceptance behaviors. To address these concerns, we utilized an instrumental-variable method based on gatekeepers' review habits, which is in keeping with the approach used in Hegde, Ljungqvist, and Raj (2021). Specifically, we measured a gatekeeper's knowledge sharing habit using the average number of knowledge sharing comments in pull requests previously reviewed by the gatekeeper, and measured gatekeeping leniency using the average acceptance rate in pull requests previously reviewed by the gatekeeper. We then used the two-stage residual inclusion (TSRI) model to include our instrumental variables in the regression, since the TSRI model is generally statistically consistent in the class of nonlinear models (Terza, Basu, & Rathouz, 2008). We first regressed the endogenous variables (*Knowledge Sharing* and *Code Acceptance*) on the two instrumental variables, and then used the residuals from the first stage regressions as additional controls in our cox regression.

There exist two main challenges to our IV approach: 1) these two instrumental variables should be independently related to each other; and 2) the assignment of newcomers' pull requests to gatekeepers with various knowledge sharing habits and gatekeeping leniency needs to be random. To overcome the first challenge, we plotted the distribution of knowledge sharing habit and gatekeeping leniency in Figure 1, and found a low correlation between these two variables. We also performed an OLS regression between the instrumental variables and the slope coefficient is insignificant (p=0.169), which indicates the independence between a gatekeeper's knowledge sharing habit and gatekeeping leniency.

----------Insert Figure 1 about here----------

Regarding the assignment of newcomers' pull requests, we argue that such assignment is likely to be at least semi-random, because 1) each repository has on average more than 4 gatekeepers, making it difficult for newcomers to target a particular gatekeeper; and 2) GitHub enables automatic assignment of gatekeepers based on gatekeepers' workload and availability. To further ensure the appropriateness of our IV approach, we excluded newcomers whose pull requests were assigned to gatekeepers with fewer than 10 prior reviews. We also excluded newcomers who initially participated in repositories with fewer than 2 gatekeepers, and were left with a sample of 9,271 newcomers. In addition, we implemented a validation test (Righi & Simcoe, 2019) to examine the random assignment assumption. This test suggests that the coefficients for the instrumental variables in the first stage regressions should be invariant to the characteristics of newcomer, pull request, repository, and gatekeeper, and therefore including more controls should not significantly affect these coefficients. To conduct this test, we included all newcomer level, pull request level, repository level, and gatekeeper level controls that emerge before the pull request assignment in the first stage regressions (Table 2). The results in Table 2 suggest that the instrumental variables are strong in predicting gatekeepers' knowledge sharing and code acceptance activities. In addition, the coefficients for the instrumental variables do not change significantly after adding more controls, which indicates that both instruments pass the validation test (Hedge et al., 2021).

----------Insert Table 2 about here----------

RESULTS

Table 3 provides descriptive statistics and correlation coefficients for all variables. To test for multicollinearity, we also checked the variance inflation factor (VIF) for each variable. All VIF values were well below the threshold of 10 (ranging from 1.00 to 3.64), suggesting no multicollinearity in our regression analysis.

----------Insert Table 3 about here----------

The results of our main analysis are presented in Table 4. Column 1 is the baseline Cox regression model. Column 2 includes all newcomer level, pull request level, repository level and gatekeeper level controls. Column 3 is the Cox regression model using the TSRI method, and column 4 is the adjusted Cox regression model with controls. These results support H1 and H2, which predict that knowledge sharing and code acceptance will positively influence newcomers' continued contribution. The coefficients of *Knowledge Sharing* and *Code Acceptance* are positive and significant in all models ($p<0.005$), which highlights the important role of gatekeepers in affecting newcomers' willingness to stay in the OSS projects.

----------Insert Table 4 about here----------

H3a proposes that the influence of gatekeepers' knowledge sharing on newcomers' continued participation will be magnified if gatekeepers have high expertise. This hypothesis is supported given the positive coefficients ($p<0.05$) on the interaction terms between knowledge sharing and gatekeepers' expertise in column 1 and column 2. These effects become more significant ($p<0.005$) after the inclusion of TSRI residuals in column 3 and column 4. Similarly, H3b is also supported since we do not observe significant coefficients on the interaction terms between code acceptance and gatekeepers' expertise in any of the models.

H4a and H4b suggest that both gatekeepers' knowledge sharing and code acceptance will have weaker influences on newcomers' continued participation if the newcomers work for the OSS firm. Although the coefficients on the interaction terms (*Knowledge sharing* × *Newcomer affiliation*) and (*Code acceptance* × *Newcomer affiliation*) are negative, they are statistically insignificant ($p>0.10$), and therefore H4a and H4b are not supported. These findings have two main implications: 1) Newcomers who work for the OSS firm can also garner private benefits from gatekeepers' knowledge sharing and code implementation (perhaps because they want to exert influence on the OSS projects); and 2) Newcomers who work for the OSS firm may not have extra private information about the projects they initially participate in if they are not the maintainers of these projects, and they also rely on gatekeepers' activities to form future beliefs.

Supplementary analysis and robustness test

Supplementary analysis

The above results have suggested that gatekeepers' knowledge sharing and code acceptance behaviors have positive effects on newcomers' motivations to continue their participation in the same OSS project. Another important question is whether gatekeepers' behaviors will affect newcomers' overall beliefs on the OSS projects owned by the firm, instead of just the project they initially participate in. To explore such effects, we conducted a supplementary analysis by redefining the dependent variable as a dummy that takes the value of 1 if the newcomer submits a second pull request to any OSS projects owned by the firm following his initial contribution. The results in Table 5 suggest that both gatekeepers' knowledge sharing and code acceptance behaviors positively affect newcomers' continued participation in projects owned by the OSS firm, as shown by the positive coefficients ($p<0.005$) on *Knowledge Sharing* and *Code Acceptance* in all models. This indicates that newcomers rely on their initial participation experiences to form expectations about other OSS projects owned by the firm as well. In addition, the negative and significant interaction terms (*Knowledge sharing* × *Newcomer affiliation*) and (*Code acceptance* × *Newcomer affiliation*) suggest that newcomers who work for the OSS firm rely less on their initial participation experiences to make continued participation decisions in firm-hosted OSS projects. This can make intuitive sense because employee newcomers can get access to private information about at least some of the firm-hosted OSS projects (e.g., offline interactions with the project maintainers), and therefore depend less on their initial participation experiences to form expectations about these projects.

----------Insert Table 5 about here----------

Robustness tests

We conducted several robustness checks to ensure the validity of our results. First, an implicit prediction of our model is that contributors' continued participation will be less affected if they rely less on current experiences to form future beliefs. To validate this, we performed an alternative analysis using the sample of established contributors who have made at least one contribution to the OSS project before their current participation, since established contributors have obtained private information regarding gatekeepers' preferences via prior interactions, and therefore depend less on current participation to update their beliefs. The results are presented in Table 6. Columns 1 through 3 display the endogeneity-adjusted regression results using the sample of established contributors who have submitted one, two, and three pull requests before the focal submissions. Consistent with the prediction, we found that the effects of gatekeepers' knowledge sharing and code acceptance behaviors on continued participation are weakened as contributors increasingly participate in the OSS projects.

----------Insert Table 6 about here----------

Second, given the possibility that using the deep learning classifier to measure knowledge sharing activities may be subject to false-negative errors, we reconstructed this variable by computing the total number of comments given by a gatekeeper in a pull request (see column 1 in Table 7). Third, we examined the influence of a specific type of knowledge on contributors' continued participation by focusing on comments on code formatting (e.g., naming, code comments, and indentation). Sharing knowledge about code formatting plays an important role in improving the readability of contributors' submitted code and maintaining the consistency of the overall OSS codebase (Sadowski et al., 2018). To do so, we utilized Google's code style guides (Hartman, 2017) to extract a list of words related to code formatting, and then performed a keyword matching to capture knowledge sharing comments (see column 2 in Table 7). Forth, instead of using the TSRI method to solve the endogeneity problem, we performed a two-stage least squares estimation with a linear probability model in the second stage (Table 7, column 3). As shown in Table 7, all three robustness tests provide results that are consistent with our main analysis.

----------Insert Table 7 about here----------

DISCUSSION AND CONCLUSION

The success of firms' participation in OSS development significantly depends upon their ability to retain existing contributors in their OSS projects. In this study, we focus on how gatekeepers' knowledge sharing and code acceptance activities will influence newcomers' continued participation in firm-hosted OSS projects. To address the endogeneity concern, we employ an instrumental-variable method that utilizes the variation in gatekeepers' review patterns. We find that newcomers will be motivated to continue their participation if they receive knowledge from gatekeepers or if their code is accepted. These effects hold true among newcomers who work for the OSS firm. Meanwhile, the positive influence of gatekeepers' knowledge sharing would be magnified if gatekeepers have high expertise. We also find evidence that gatekeepers' interactions will affect newcomers' beliefs on the entire pool of OSS projects owned by the firm, instead of just the project they initially participate in.

Our study makes three contributions to the literature. First, we contribute to research on contributors' continued participation (Dejean et al., 2015; Fang et al., 2009; Piezunka, & Dahlander, 2019) in firm-hosted OSS projects by focusing on the critical role of gatekeepers, the people representing the OSS firm. We show that gatekeepers' interactions with newcomers may significantly influence continued participation via affecting newcomers' perceived benefits, and that such influence may persist across and within organizational boundaries.

Second, we provide a belief-updating perspective on how prior interactions will affect contributors' future participation. Although existing research on continued participation has paid attention to the importance of contributors' prior participation (Dejean et al., 2015), the specific mechanism by which prior participation shapes contributors' willingness to stay in OSS projects is still ambiguous. By utilizing the belief updating framework, we are able to get a clearer picture of how contributors, as rational decision makers, form continued participation decisions by taking into account their prior experiences of participation. Specifically, we show that contributors' expectations about the benefit of OSS participation will be updated based on the new information they receive from interactions with gatekeepers, and that such belief updating process relies crucially on contributors' preferences over OSS participation as well as on the accessibility of private information before contributors' subsequent participation.

Third, we also contribute to research and practice on OSS project governance (He, Puranam, Shrestha, & von Krogh, 2020; O'Mahony & Karp, 2020) by emphasizing the importance for firms to regulate and guide gatekeepers' behaviors. The significant influence of gatekeepers' knowledge sharing and code acceptance activities on newcomers' continued participation suggests that gatekeepers not only play a role in managing and monitoring voluntary contributions, but also comprise an important strategic asset that assists OSS firms in sustaining a constant flow of voluntary contributions and pursuing long-term success. As a result, firms should strategically manage the behaviors of their gatekeepers so that contributors can form good impressions about the OSS projects and have more incentives to continue their participation. This is important especially when contributors' initial participation experiences affect their overall beliefs about all the OSS projects owned by the firm.

As with all studies, our study has several limitations that may provide opportunities for future research. First, our study only focuses on the public interactions between gatekeepers and contributors on GitHub, while contributors can contact and communicate with gatekeepers via other private means such as emails and developer conferences (Fang, Wu, & Clough, 2020). Future research can examine the influence of these communication channels. Second, our formal model assumes constant costs required by contributors to complete their work, while in reality contributors are able to strategically determine their effort levels based on the beliefs (Belenzon, & Schankerman, 2015; Foss, Frederiksen, & Rullani, 2016). As a result, future work can study how contributors adjust their effort levels in the belief updating process, and how such adjustments will affect the OSS firm's benefits. It may be even more interesting for future studies to model the strategic behaviors of gatekeepers given the rational decisions made by contributors (e.g., by utilizing a principal-agency model) and answer questions such as "to whom will gatekeepers share knowledge to elicit high effort levels and continued participation?" Third, while we recognize that the continued participation decision is subjective and depends on the characteristics of contributors and gatekeepers, we only examine how contributors' organizational affiliation and gatekeepers' expertise will affect continued participation, and future research can explore the influence of other characteristics of contributors and gatekeepers. Forth, although we find evidence that knowledge sharing and code acceptance will motivate contributors to stay in the OSS projects, we are not in a position to tell how this increase in participation will affect the overall quality of the projects. For instance, while sharing knowledge may help improve contributors' code quality in the subsequent participation, a high code acceptance rate may result in an inflow of low-quality submissions (Chen et al., 2022; Kretschmer, Leiponen, Schilling, & Vasudeva, 2020). Therefore, future studies can analyze the tradeoffs between motivating continued participation and ensuring the code quality of the firm-hosted OSS projects.

REFERENCES

Acar, O. A. (2019). Motivations and solution appropriateness in crowdsourcing challenges for innovation. Research Policy. 48(8), 103716.

Alexy, O., West, J., Klapper, H., and Reitzig, M. (2018). Surrendering control to gain advantage: Reconciling openness and the resource-based view of the firm. Strategic Management Journal. 39(6), 1704-1727.

Allen, T. J., and Cohen, S. I. (1969). Information Flow in Research and Development Laboratories. Administrative Science Quarterly. 14(1), 12-19.

Ambuehl, S., and Li, S. (2018). Belief updating and the demand for information. Games and Economic Behavior. 109, 21-39.

Asay, M. (2018). Who really contributes to open source. https://www.infoworld.com/article/3253948/who-really-contributes-to-open-source.html

Augenblick, N., and Rabin, M. (2021). Belief Movement, Uncertainty Reduction, and Rational Updating. The Quarterly Journal of Economics. 136(2), 933–985.

Baldwin, C. Y., and Clark, K. B. (2006). The architecture of participation: Does code architecture mitigate free riding in the open source development model? Management Science. 52(7), 1116–1127.

Bartol, K. M., and Srivastava, A. (2002). Encouraging Knowledge Sharing: The Role of Organizational Reward Systems. Journal of Leadership & Organizational Studies. 9(1), 64-76.

Benjamin, D. J. (2019). Handbook of Behavioral Economics: Applications and Foundations 1.

Belenzon, S., and Schankerman, M. (2015). Motivation and sorting of human capital in open innovation. Strategic Management Journal. 36(6), 795-820.

Camerer, C. F., and Loewenstein, G. (2004). Advances in Behavioral Economics. Princeton University Press.

Charness, G., and Levin, D. (2005). When Optimal Choices Feel Wrong: A Laboratory Study of Bayesian Updating, Complexity, and Affect. American Economic Review. 95(4), 1300-1309.

Chen, L., Tong, T. W., Tang, S., and Han, N. (2022). Governance and Design of Digital Platforms: A Review and Future Research Directions on a Meta-Organization. Journal of Management. 48(1), 147-184.

Dahlander, L., and Magnusson, M. G. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. Research Policy. 34(4), 481–493.

Dahlander, L., and O'Mahony, S. (2011). Progressing to the center: Coordinating project work. Organization Science. 22(4), 961–979.

Daniel, S., Maruping, L. M., Cataldo, M., and Herbsleb, J. (2018). The Impact of Ideology Misfit on Open Source Software Communities and Companies. MIS Quarterly. 42(4), 1069-1096.

Dejean, S., and Jullien, N. (2015). Big from the beginning: Assessing online contributors' behavior by their first contribution. Research Policy. 44(6), 1226-1239.

Fang, T. P., Wu, A., and Clough, D. R. (2020). Platform diffusion at temporary gatherings: Social coordination and ecosystem emergence. Strategic Management Journal. In Press.

Fang, Y., and Neufeld, D. (2009). Understanding Sustained Participation in Open Source Software Projects. Journal of Management Information Systems. 25(4), 9-50.

Hann, I. H., Roberts, J. A., and Slaughter, S. A. (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), 520–538.

He, V. F., Puranam, P., Shrestha, Y. R., and von Krogh, G. (2020). Resolving governance disputes in communities: A study of software license decisions. Strategic Management Journal. 41(10), 1837-1868.

Hegde, D., A. Ljungqvist, and M. Raj (2021) Quick or Broad Patents? Evidence from U.S. Startups. Forthcoming at Review of Financial Studies.

Ho, S.Y., and Rai, A. (2017). Continued voluntary participation intention in firm-participating open source software projects. Information Systems Research. 28(3), 603–625.

Jeppesen, L. B., and Frederiksen, L. (2006). Why Do Users Contribute to Firm-Hosted User Communities? The Case of Computer-Controlled Music Instruments. Organization Science. 17(1), 45-63.

Kammer, L., Hodges, M., and Murillo, A. (2019). Code Health: Respectful Reviews == Useful Reviews. https://testing.googleblog.com/2019/11/code-health-respectful-reviews-useful.html

Karahanna, E., Straub, D. W., and Chervany, N. L. (1999). Information technology adoption across time: A cross-sectional comparison of preadoption and post-adoption beliefs. MIS Quarterly. 23(2), 183–213.

Kim, K., and Viswanathan, S. (2019). The experts in the crowd: The role of experienced investors in a crowdfunding market. MIS Quarterly.

Kim, S. N., Cavedon, L., and Baldwin, T. (2010). Classifying Dialogue Acts in One-on-one Live Chats. Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, 862–871.

Kretschmer, T., Leiponen, A., Schilling, M., and Vasudeva, G. (2020). Platform ecosystems as metaorganizations: Implications for platform strategies. Strategic Management Journal. In Press.

Lochner, L. (2007). Individual Perceptions of the Criminal Justice System. American Economic Review. 97(1), 444-460.

Massa, F. G., and O'Mahony, S. (2021). Order from Chaos: How Networked Activists Self-Organize by Creating a Participation Architecture. Administrative Science Quarterly. 66(4), 1037–1083.

Morrison, E. W. (1993). Newcomer Information Seeking: Exploring Types, Modes, Sources, and Outcomes. Academy of Management Journal. 36(3), 557-589.

O'Mahony, S., and Karp, R. (2020). From proprietary to collective governance: How do platform participation strategies evolve? Strategic Management Journal. In Press.

Piezunka, H., and Dahlander, L. (2019). Idea Rejected, Tie Formed: Organizations' Feedback on Crowdsourced Ideas. Academy of Management Journal. 62(2), 503–530.

Riedl, C., and Seidel, P. V. (2018). Learning from Mixed Signals in Online Innovation Communities. Organization Science. 29(6), 989-1236.

Righi, C., and Simcoe, T. (2019). Patent examiner specialization. Research Policy. 48(1), 137-148.

Sadowski, C., Söderberg, E., Church, L., Sipko, M., and Bacchelli, A. (2018). Modern Code Review: A Case Study at Google. ACM/IEEE 40th International Conference on Software Engineering: Software Engineering in Practice. 181(2), 181-190.

Scharfstein, D. S., and Stein, J. C. (1990). Herd Behavior and Investment. American Economic Review. 80(3), 465-479.

Shah, S. K. (2006). Motivation, governance and the viability of hybrid forms in open source software development. Management Science. 52(7), 1000–1014.

Singh, P. V., Tan, Y., and Youn, N. (2011). A Hidden Markov Model of Developer Learning Dynamics in Open Source Software Projects. Information Systems Research. 22(4), 685-891.

Spaeth, S., von Krogh, G., and He, F. (2015). Perceived firm attributes and intrinsic motivation in sponsored open source software projects. Information Systems Research. 26(1), 224–237.

Terza, J. V., Basu, A., and Rathouz, P. J. (2008). Two-stage residual inclusion estimation: Addressing endogeneity in health econometric modeling. Journal of Health Economics. 27(3), 531-543.

von Krogh, G., Haefliger, S., Spaeth, S., and Wallin, M. W. (2012). Carrots and Rainbows: Motivation and Social Practice in Open Source Software Development. MIS Quarterly. 36(2), 649-676.

von Krogh, G., Spaeth, S., and Lakhani, K.R. (2003). Community, joining, and specialization in open source software innovation: A case study. Research Policy. 32(7), 1217–1241.

von Hippel, E., and von Krogh, G. (2003). Open source software and the "private-collective" innovation model: Issues for organization science. Organization Science. 14(2), 209–

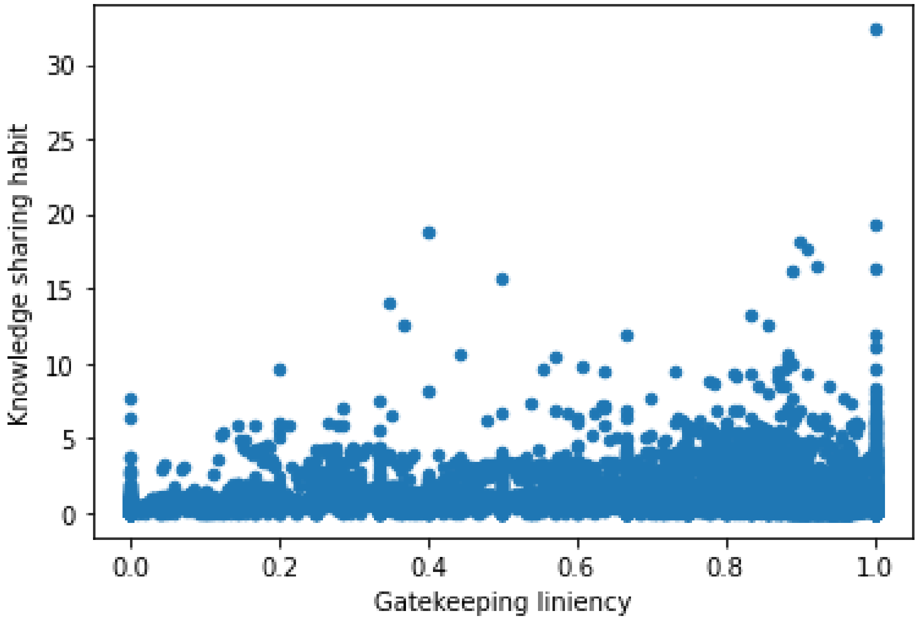
West, J. (2003). How open is open enough? Melding proprietary and open source platform strategies. Research Policy. 32(7), 1259–1285.

Zaggl, M. A. (2017). Manipulation of explicit reputation in innovation and knowledge exchange communities: The example of referencing in science. Research Policy. 46(5), 970-983.

Zimmermann, F. (2020). The Dynamics of Motivated Beliefs. American Economic Review. 110(2), 337-361.

Zhang, C., Hahn, J., and De, P. (2013). Continued participation in online innovation communities: Does community response matter equally for everyone? Information Systems Research. 24(4), 1112–1130.

Figure 1. Distribution of gatekeeping leniency and knowledge sharing habit



Note. Figure 1 shows the distribution of knowledge sharing habit and gatekeeping leniency. Knowledge sharing habit is defined as the average number of knowledge sharing comments in pull requests previously reviewed by the gatekeeper. Gatekeeping leniency is defined as the average acceptance rate in pull requests previously reviewed by the gatekeeper.

Table 1. Examples of labeled utterances

Label	Utterance
Acknowledge	Yes please.
Action-directive	Please resolve conflicts.
Appreciation	Good catch!
Collaborative Completion	Merging.
Offers, Options Commits	Let me know if you have any thoughts.
Apology	Sorry about that.
Declarative Yes-No-Question	Copying a subset of api.json?
Open-Question	What do you think?
Statement-non-opinion	I'll try and take a look today or tomorrow.
Statement-opinion	This change is insufficient for the desired effect.
Yes-No-Question	Is this duplicated with PR #199?
Wh-Question	What is this map doing?
Hedge	I'm not sure.

Note. Table 1 shows examples of labels assigned to an utterance using the dialogue act classifier.

Table 2. First Stage Regressions

	Knowledge Sharing		Code Acceptance	
	(1)	**(2)**	**(3)**	**(4)**
Knowledge sharing IV	0.183*	0.189*	0.001	0.001
	(0.080)	(0.082)	(0.011)	(0.010)
Code acceptance IV	0.412	0.409	0.538***	0.536***
	(0.288)	(0.295)	(0.076)	(0.074)
Gatekeeper expertise		0.002		-0.001*
		(0.004)		(0.000)
Newcomer repo		-0.000***		0.000
		(0.000)		(0.000)
Newcomer age		-0.002+		0.001***
		(0.001)		(0.000)
Newcomer star given		-0.000**		-0.000
		(0.000)		(0.000)
Description length		0.000		0.000
		(0.000)		(0.000)
Pull request commit		0.012		-0.001
		(0.009)		(0.001)
Code change		0.000		0.000*
		(0.000)		(0.000)
Gatekeeper experience		-0.000		-0.000
		(0.000)		(0.000)
Repo fork received		-0.000		0.000
		(0.000)		(0.000)
Repo pull request received		0.000		-0.000
		(0.000)		(0.000)
Repo age		0.001		-0.000
		(0.002)		(0.000)
Affiliation		0.017		0.072***
		(0.102)		(0.013)
Repository-by-year FE	Yes	Yes	Yes	Yes

Note. This table reports the first stage regression results. All continuous variables are normalized. All models include a full set of repository-by-year fixed effects given the assumption that pull requests are assigned to gatekeepers randomly within a repository and year. Significance levels: + p<0.1, * p<0.05, ** p<0.01, *** p<0.005.

Table 3. Descriptive Statistics and Correlations

No	Variable	Mean	SD	Min	Max	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	Continued participation	0.01	0.08	0	1.00	1.00																	
2	Knowledge sharing	0.85	2.49	0	113.00	0.04	1.00																
3	Code acceptance	0.67	0.47	0	1.00	0.03	-0.06	1.00															
4	Newcomer affiliation	0.13	0.33	0	1.00	0.02	0.00	0.11	1.00														
5	Gatekeeper expertise	148.78	1082.92	0	20030.00	0.00	-0.02	-0.02	-0.03	1.00													
6	Newcomer star given	171.31	680.25	0	30743.00	-0.01	-0.02	0.02	-0.05	0.02	1.00												
7	Newcomer repo	49.05	208.24	0	22618.00	-0.01	-0.01	0.01	-0.02	0.00	0.30	1.00											
8	Newcomer age	53.77	31.59	1	164.00	0.02	-0.01	0.17	0.08	-0.02	0.09	0.06	1.00										
9	Repo pull request received	671.58	1134.74	0	18137.00	-0.01	-0.02	0.03	0.02	0.00	-0.01	-0.01	0.06	1.00									
10	Repo fork received	1037.32	1731.05	0	21926.00	-0.01	-0.02	-0.03	-0.02	0.00	-0.02	-0.01	0.02	0.34	1.00								
11	Direct edit	0.03	0.17	0	1.00	0.01	0.04	0.05	0.02	-0.02	0.01	0.00	0.03	0.01	0.01	1.00							
12	Repo age	28.84	20.26	1	105.00	0.02	0.02	0.00	-0.03	-0.02	-0.01	-0.02	0.23	0.09	0.08	0.04	1.00						
13	Review time	68.62	210.68	0	2210.81	-0.02	0.01	-0.30	-0.01	-0.03	-0.01	-0.02	-0.06	-0.02	0.01	0.03	-0.04	1.00					
14	Code change	483.48	17850.05	0	1338017.00	0.00	0.00	0.00	0.00	0.00	-0.01	0.00	-0.01	0.01	0.03	0.00	-0.01	0.00	1.00				
15	Pull request commit	1.96	7.17	0	951.00	0.01	0.07	-0.10	-0.03	0.01	-0.02	-0.01	-0.07	-0.01	-0.01	0.11	0.01	0.05	0.01	1.00			
16	Description length	32.54	84.80	0	21696.00	0.02	0.10	-0.02	0.00	-0.01	-0.02	0.00	0.04	0.02	0.00	0.02	0.08	0.02	0.00	0.06	1.00		
17	External comment	0.50	1.84	0	66.00	0.01	0.21	-0.02	-0.01	-0.02	-0.01	0.00	0.02	-0.02	-0.01	0.04	0.06	0.09	0.02	0.04	0.08	1.00	
18	Gatekeeper experience	123.84	230.99	10	2860.00	0.02	-0.03	0.06	0.02	-0.01	0.01	0.00	0.10	0.08	0.04	0.07	0.36	-0.08	-0.01	0.00	0.03	0.05	1.00

Table 4. Main Results

	COX		TSRI	
	(1)	(2)	(3)	(4)
Knowledge sharing	0.041*** (0.005)	0.038*** (0.006)	0.070*** (0.013)	0.091*** (0.015)
Code acceptance	0.709*** (0.066)	0.596*** (0.069)	1.003*** (0.102)	0.897*** (0.118)
Gatekeeper expertise	-0.01 (0.012)	-0.012 (0.015)	-0.011 (0.012)	-0.012 (0.014)
Knowledge sharing × Gatekeeper expertise	0.001* (0.000)	0.001* (0.000)	0.001*** (0.000)	0.001*** (0.000)
Code acceptance × Gatekeeper expertise	0.009 (0.012)	0.011 (0.015)	0.011 (0.012)	0.012 (0.014)
Newcomer affiliation	0.632*** (0.161)	0.585*** (0.162)	0.621*** (0.161)	0.567*** (0.173)
Knowledge sharing × Newcomer affiliation	-0.007 (0.012)	-0.005 (0.012)	-0.007 (0.012)	-0.018 (0.013)
Code acceptance × Newcomer affiliation	-0.231 (0.171)	-0.223 (0.171)	-0.230 (0.171)	-0.212 (0.183)
Knowledge sharing (residual)			-0.033* (0.015)	-0.056*** (0.016)
Code acceptance (residual)			-0.463*** (0.116)	-0.443*** (0.130)
Direct edit		0.030 (0.112)		-0.002 (0.122)
Repo fork received		0.000 (0.000)		0.000 (0.000)
Repo pull request received		0.000 (0.000)		0.000 (0.000)
Newcomer Star Given		-0.000* (0.000)		-0.000+ (0.000)
Newcomer repo		-0.000 (0.000)		-0.000 (0.000)
Repo age		-0.006*** (0.001)		-0.007*** (0.001)
Newcomer age		0.001* (0.001)		0.002* (0.001)
Review time		-0.001*** (0.000)		-0.001*** (0.000)
Code change		-0.000 (0.000)		-0.000 (0.000)
Pull request commit		0.003*** (0.000)		0.003*** (0.000)
Description length		0.000*** (0.000)		0.000*** (0.000)
External comment		0.020*** (0.007)		0.017* (0.007)
Gatekeeper experience		0.000*** (0.000)		0.000*** (0.000)

Note. Standard errors are clustered at the newcomer level given the fact that newcomers have multiple monthly events. Significance levels: + p<0.1, * p<0.05, ** p<0.01, *** p<0.005.

Table 5. Supplementary Analysis

	COX		TSRI	
	(1)	(2)	(3)	(4)
Knowledge sharing	0.039***	0.037***	0.096***	0.089***
	(0.006)	(0.006)	(0.013)	(0.013)
Code acceptance	0.556***	0.469***	0.793***	0.664***
	(0.059)	(0.063)	(0.094)	(0.099)
Gatekeeper expertise	-0.012	-0.015	- 0.012	-0.014
	(0.010)	(0.013)	(0.010)	(0.012)
Knowledge sharing × Gatekeeper expertise	0.001***	0.001***	0.001***	0.001***
	(0.000)	(0.000)	(0.000)	(0.000)
Code acceptance × Gatekeeper expertise	0.010	0.013	0.011	0.012
	(0.010)	(0.013)	(0.010)	(0.012)
Newcomer affiliation	1.04***	1.003***	1.042***	1.012***
	(0.118)	(0.119)	(0.119)	(0.121)
Knowledge sharing × Newcomer affiliation	-0.023+	-0.024*	-0.027*	-0.029*
	(0.012)	(0.012)	(0.011)	(0.011)
Code acceptance × Newcomer affiliation	-0.403***	-0.383***	-0.414***	-0.396***
	(0.128)	(0.129)	(0.129)	(0.130)
Knowledge sharing (residual)			-0.062***	-0.056***
			(0.014)	(0.014)
Code acceptance (residual)			-0.359***	-0.278***
			(0.108)	(0.108)
Direct edit		-0.123		-0.114
		(0.112)		(0.112)
Repo fork received		-0.000+		-0.000+
		(0.000)		(0.000)
Repo pull request received		0.000*		0.000*
		(0.000)		(0.000)
Newcomer Star Given		0.000*		0.000*
		(0.000)		(0.000)
Newcomer repo		0.000		0.000
		(0.000)		(0.000)
Repo age		-0.006***		-0.006***
		(0.001)		(0.001)
Newcomer age		0.002*		0.002*
		(0.001)		(0.001)
Review time		-0.001***		-0.001***
		(0.000)		(0.000)
Code change		0.000		0.000
		(0.000)		(0.000)
Pull request commit		0.002***		0.002***
		(0.000)		(0.000)
Description length		0.000***		0.000***
		(0.000)		(0.000)
External comment		0.020***		0.016*
		(0.007)		(0.006)
Gatekeeper experience		0.000**		0.000**
		(0.000)		(0.000)

Note. Standard errors are clustered at the newcomer level given the fact that newcomers have multiple monthly events. Significance levels: + p<0.1, * p<0.05, ** p<0.01, *** p<0.005.

Table 6. Robustness Test: Regression Results for Established Contributors

	TSRI		
	One prior pull request (1)	**Two prior pull requests (2)**	**Three prior pull requests (3)**
Knowledge sharing	0.023	0.027	0.051
	(0.025)	(0.035)	(0.034)
Code acceptance	0.930***	0.468	0.385
	(0.255)	(0.332)	(0.341)
Gatekeeper expertise	-0.042	- 0.121	-0.087
	(0.040)	(0.097)	(0.060)
Knowledge sharing × Gatekeeper expertise	0.001***	0.001*	0.001
	(0.000)	(0.000)	(0.002)
Code acceptance × Gatekeeper expertise	0.039	0.116	0.088
	(0.040)	(0.097)	(0.060)
Newcomer affiliation	0.232	-0.383	0.088
	(0.353)	(0.623)	(0.533)
Knowledge sharing × Newcomer affiliation	0.009	0.004	0.007
	(0.019)	(0.024)	(0.045)
Code acceptance × Newcomer affiliation	0.006	0.430	-0.064
	(0.363)	(0.636)	(0.548)
Knowledge sharing (residual)	-0.014	0.004	-0.049
	(0.025)	(0.036)	(0.035)
Code acceptance (residual)	-0.895***	-0.522	-0.582
	(0.273)	(0.365)	(0.368)
Direct edit	0.038	-0.120	0.033
	(0.160)	(0.284)	(0.252)
Repo fork received	-0.000	0.000	-0.000
	(0.000)	(0.000)	(0.000)
Repo pull request received	-0.000	-0.000*	-0.000
	(0.000)	(0.000)	(0.000)
Newcomer Star Given	-0.000	0.000	-0.000
	(0.000)	(0.000)	(0.000)
Newcomer repo	-0.001+	-0.001	0.001
	(0.001)	(0.001)	(0.001)
Repo age	-0.004*	-0.004	-0.005+
	(0.002)	(0.003)	(0.003)
Newcomer age	-0.002+	-0.001	-0.001
	(0.001)	(0.002)	(0.002)
Review time	-0.002**	-0.018***	-0.002
	(0.001)	(0.006)	(0.001)
Code change	-0.000	-0.000+	-0.000
	(0.000)	(0.000)	(0.000)
Pull request commit	0.002***	0.002***	0.001*
	(0.000)	(0.001)	(0.001)
Description length	-0.000	-0.000	-0.001
	(0.000)	(0.001)	(0.001)
External comment	0.004**	0.002+	-0.004
	(0.002)	(0.001)	(0.006)
Gatekeeper experience	0.000*	0.000*	-0.000
	(0.000)	(0.000)	(0.000)

Note. Standard errors are clustered at the contributor level given the fact that contributors have multiple monthly events. Significance levels: + p<0.1, * p<0.05, ** p<0.01, *** p<0.005.

Table 7. Robustness Test: alternative independent variables and instrumental variable analyses

	General comments	Code-formatting knowledge sharing	TSLS
	(1)	(2)	(3)
Knowledge sharing	0.036***	0.156***	0.021***
	(0.005)	(0.030)	(0.004)
Code acceptance	0.897***	0.890***	0.131***
	(0.119)	(0.119)	(0.018)
Gatekeeper expertise	-0.012	- 0.012	-0.004+
	(0.014)	(0.014)	(0.002)
Knowledge sharing × Gatekeeper expertise	0.001***	0.004*	0.002*
	(0.000)	(0.002)	(0.001)
Code acceptance × Gatekeeper expertise	0.011	0.012	0.003
	(0.014)	(0.014)	(0.003)
Newcomer affiliation	0.603***	0.561***	0.067
	(0.174)	(0.174)	(0.046)
Knowledge sharing × Newcomer affiliation	-0.008+	-0.048*	-0.010
	(0.005)	(0.024)	(0.012)
Code acceptance × Newcomer affiliation	-0.234	-0.210	0.044
	(0.184)	(0.184)	(0.055)
Knowledge sharing (residual)	-0.022***	-0.085**	
	(0.006)	(0.031)	
Code acceptance (residual)	-0.417***	-0.442***	
	(0.131)	(0.131)	
Direct edit	-0.017	-0.004	0.003
	(0.123)	(0.122)	(0.025)
Repo fork received	0.000	0.000	-0.000
	(0.000)	(0.000)	(0.000)
Repo pull request received	0.000	-0.000	-0.000
	(0.000)	(0.000)	(0.000)
Newcomer Star Given	-0.000+	-0.000+	-0.000*
	(0.000)	(0.000)	(0.000)
Newcomer repo	-0.000	-0.000	-0.000***
	(0.000)	(0.000)	(0.000)
Repo age	-0.007***	-0.007***	-0.001***
	(0.001)	(0.001)	(0.000)
Newcomer age	0.001+	0.001+	0.000*
	(0.001)	(0.001)	(0.000)
Review time	-0.001***	-0.001***	-0.000***
	(0.000)	(0.000)	(0.000)
Code change	-0.000	-0.000	-0.000
	(0.000)	(0.000)	(0.000)
Pull request commit	0.003***	0.003***	0.001***
	(0.000)	(0.000)	(0.000)
Description length	0.000***	0.000***	0.000***
	(0.000)	(0.000)	(0.000)
External comment	0.018*	0.020*	0.011***
	(0.007)	(0.008)	(0.003)
Gatekeeper experience	0.000***	0.000***	0.000***
	(0.000)	(0.000)	(0.000)

Note. Standard errors are clustered at the contributor level given the fact that contributors have multiple monthly events. Significance levels: + $p<0.1$, * $p<0.05$, ** $p<0.01$, *** $p<0.005$.