Platform Governance Matters:
How Platform Gatekeeping Affects Knowledge Sharing among Complementors

Yuchen Zhang
Freeman School of Business
Tulane University
7 McAlister Dr., New Orleans, LA 70118
Tel.: (504) 314-2467
E-mail: yzhang54@tulane.edu

Jingjing Li
McIntire School of Commerce
University of Virginia
125 Ruppel Dr., Charlottesville, VA 22903
Tel.: (434) 924-8981
E-mail: jl9rf@comm.virginia.edu

Tony W. Tong*
Leeds School of Business
University of Colorado Boulder
995 Regent Dr., Boulder, CO 80309
Tel.: (303) 492-0141
Email: tony.tong@colorado.edu

February 2018

* Corresponding author.

Platform Governance Matters:
How Platform Gatekeeping Affects Knowledge Sharing among Complementors

ABSTRACT

Complementors are critical to the success of a platform, yet complementor development proves challenging. Emerging research suggests that platform owners can design governance policies to shape complementors' behaviors, but current theory development is still in a nascent stage. We contribute to this stream of research by empirically examining the relationship between gatekeeping, a prominent platform governance policy, and complementors' knowledge sharing activity. By exploiting the jailbreak of iOS 7 as an exogenous shock to Apple's gatekeeping policy and tracing iOS and Android app developers' knowledge sharing activity on two online forums, we find causal evidence that a deficiency in gatekeeping reduces the quantity and quality of knowledge sharing. We discuss two potential mechanisms, market competition and community collaboration, and their theoretical implications.

Keywords: Platform governance, platform owner, complementor, gatekeeping, knowledge sharing, innovation

INTRODUCTION

Complementors are critical to the success of a two-sided platform (McIntyre & Srinivasan, 2017; Parker & Van Alstyne, 2005; Wareham, Fox, & Giner, 2014). More high-quality complementors can provide users with a larger number and variety of product offerings, enhancing the attractiveness of the platform. For example, Apple's App Store and Google's Play Store are more attractive to mobile device users than Windows Phone, because they offer a wider selection of apps developed by millions of active developers (Gronli *et al.*, 2014). While complementors are important, how to develop them and shape their behaviors in favor of the platform proves challenging to platform owners for at least two reasons (Yoffie & Kwak, 2006). First, it is difficult to understand and manage complementors' motivations to contribute to a platform. Complementors often have heterogeneous motivations and incentives (Wareham *et al.*, 2014), making it extremely hard to design a platform policy to accommodate and reconcile such heterogeneity. Second, platform owners have limited means at their disposal compared with traditional organizations. Conventional means such as hierarchical control and organizational incentives are less applicable in the platform context (Adner, 2017), because complementors are loosely connected to the platform and are not under the platform owner's direct control.

Scholars have long observed that platform owners can resort to design rules to shape complementors' behaviors (Baldwin & Clark, 2000), and recent research begins to explore how platform policies may affect complementors' contributions to platforms (Boudreau & Hagiu, 2009). This small, but fast developing stream of research revolves around the emergent idea of platform governance (Tiwana, 2013). Although researchers and practitioners agree on the significance of platform governance for both platform owners and complementors, strategic management research on platform governance and its implications remains rather scarce (McIntyre & Srinivasan, 2017). One important reason for the scarcity of research, we suspect, is that the concept of platform governance is still in its nascent stage and requires substantial theoretical development by scholars. Perhaps equally important is our observation that platform governance, or change in governance, is empirically difficult to identify and track, and thus scholars also face a challenge in addressing the endogeneity problem endemic in governance research more generally.

In this study, we focus on the question of how a salient aspect of platform governance, called "platform gatekeeping", may affect the knowledge sharing activity among complementors of the platform. A platform owner implements gatekeeping by adopting a set of predefined acceptance criteria that controls for what or who is allowed on the platform (Tiwana, 2013), which often produces a most immediate impact on complementors' behaviors. For example, Apple's gatekeeping policy requires apps published on its iOS platform to satisfy certain quality standards including size and interface design (Apple, 2014), which directly impact the goals, processes, and evaluations of complementors' app development activities. Moving beyond the traditional notion that gatekeeping is a screening tool (Tiwana, 2013), this study investigates how gatekeeping affects knowledge sharing among complementors. Knowledge sharing is critical to innovation-based platforms because it facilitates problem solving and solution search (Kuk, 2006; Majchrzak, Cooper, & Neece, 2004). In the context of mobile app or video game platform ecosystems (Casadesus-Masanell & Yoffie, 2007), for example, knowledge sharing can help developers quickly learn from each other's experience about product development, troubleshooting, and user preferences, all of which can lead to greater development rates and higher quality offerings. Hence, we state our research question:

How does platform gatekeeping affect knowledge sharing among complementors?

Despite the research and practical value of the question, the existing literature provides no clear theoretical guidance for the answer. On the one hand, gatekeeping can ease competitions by limiting the number of complementors; as a result, complementors may become less sensitive to knowledge protection and more willing to share. Furthermore, gatekeeping may also create a culture of exclusivity, belongingness, and collaboration among players in the community, potentially enhancing knowledge sharing to strengthen the collaborative value of the community (Cennamo & Santalo, 2013; Faraj, Jarvenpaa, & Majchrzak, 2011; Wareham *et al.*, 2014). On the other hand, gatekeeping can intensify competitions. For example, the products and services offered by the limited number of complementors may not be able to meet consumers' needs, leading to consumer exit. As a result, complementors may compete fiercely to attract consumers and limit knowledge sharing to maintain their competitive advantage (Zhu & Iansiti, 2012). Moreover, a decrease in the number of complementors may impair the development of direct and indirect reciprocity in the community, leading to a drop in knowledge sharing (Faraj & Johnson, 2011; Shah, 2006).

Given the lack of theoretical guidance, we are determined to answer this question empirically. Specifically, we adopt a quasi-experimental design and exploit the "jailbreak" of iOS 7 as a natural experiment to examine how an exogenous deficiency in Apple's gatekeeping policy affects its complementors' knowledge sharing activity. Because knowledge is embedded in individuals (Grant, 1996), we focus our analysis on knowledge sharing among app developers, which is key to the successful development of mobile apps (von Krogh & von Hippel, 2006). Studying this phenomenon of interest, however, is surrounded by two empirical challenges: first, such knowledge sharing is often taking place off the iOS platform and thus, compared to activities performed on the platform, is hard to observe; second, knowledge sharing is well known to leave no "paper trail" by which it can be measured and tracked (Krugman, 1993). To overcome these challenges, we focus on the "Internet trails", in the form of digital data, left by app developers on two online forums, StackOverflow.com and Reddit.com. To this end, we create a rich dataset on the knowledge sharing activity of about 1,200 iOS and Android app developers on the two different forums over a period of 27 weeks surrounding the jailbreak. This allows us to conduct a detailed, difference-in-differences analysis to capture the causal effects of the jailbreak on various steps of the knowledge sharing process.

This study makes several contributions to a fast-growing stream of research on strategies for platform ecosystems. First, our empirical findings advance the theoretical link between platform gatekeeping, a prominent platform governance policy, and complementors' behaviors (Adner, 2017).

Recent research suggests that, in addition to screening out inferior complementors or complements, gatekeeping can have other important implications that require investigations (Boudreau & Jeppesen, 2015). In this study, we find that gatekeeping can substantially shape complementors' knowledge sharing at the individual (app developer) level, empirically enhancing our understanding about platform gatekeeping. Importantly, we propose market competition and community collaboration as two potential theoretical mechanisms to explain our findings. Second, our study echoes prior work that managing platform network effects requires careful attention (Eisenmann, Parker, & Van Alstyne, 2006; Yoffie & Kwak, 2006). Although a larger number of complementors bring about a larger network, it may not necessarily translate into greater positive network effects, as shown by the *drop* in the quantity and quality of knowledge sharing among app developers in our study. Indeed, we find that to create and nurture network effects, platform owners need to be sensitive to both the competitive and collaborative aspects of a complementor community. Third, our study moves beyond prior research emphasis on complementors' *on*-platform activity (Boudreau & Jeppesen, 2015; Wareham *et al.*, 2014) by focusing on how complementors' *off*-platform activity responds to an exogenous shock to a platform's gatekeeping policy aimed at regulating complementors and their offerings. To the extent that complementors' off-platform activity is inherently linked to their on-platform activity such as new product launch and performance, our study highlights an important but under-researched area that can impact the overall attractiveness of the platform. Finally, our study also makes several empirical contributions. We introduce into strategic management research two novel, well-curated data sources, StackOverflow.com and Reddit.com, for empirical analysis and triangulation of results. The data sources and our empirical approach can be applied to studies of many other interesting questions about platform ecosystems, virtual communities, and knowledge management central to strategy scholars. We also go beyond econometric analysis and employ the Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003), a topic modeling method, to uncover knowledge sharing patterns among complementors and validate key assumptions of our arguments. Given its general applicability to unstructured textual data to quantify latent semantic information, LDA will prove highly valuable in future strategy research in today's increasingly digitalized economy.

RELATED LITERATURE AND RESEARCH QUESTION

Complementor development is crucial to the success of a platform, but how to manage complementors' activities proves challenging to platform owners. Complementors are heterogeneous in their motivations and capabilities to contribute value creation activities to a platform. Hence, platform owners need to strike a balance between retaining some control over complementors to ensure consistent business operations and granting them reasonable flexibility to address the heterogeneity and induce innovation. Despite such challenges, platform owners have limited means at their disposal to shape complementors, which are often not bounded by contractual obligations or hierarchical fiat, and thus are not under direct control of platform owners.

A nascent stream of research suggests platform governance as a way to address some of these challenges (Tiwana, 2013). Platform governance can significantly shape complementors' decision processes and induce value creation activities. Research has shown that well-designed governance policies can attract more complementors to join the ecosystem, induce their positive contributions, and trigger cross-side network effects, increasing the overall attractiveness of the platform (Eisenmann, Parker, & Van Alstyne, 2006). One central aspect of platform governance is control design. Control design concerns the economic and social mechanisms that platform owners can use to shape complementors' behaviors. Among various means for control design, gatekeeping, due to its importance and popularity, receives particular attention among researchers. A gatekeeping policy refers to a set of predefined acceptance criteria for judging complementors and their offerings (Tiwana, 2013). Platform owners can use gatekeeping to screen out those of a lesser value and control who and what could have access to the platform. Despite its significance for platform governance, prior research often treats gatekeeping just as a screening tool to bounce off inferior complementors or their offerings (Tiwana, 2013). However, recent research starts to observe that gatekeeping may also exert other important, nuanced influences on complementors' value creation activities. For example, Boudreau and Jeppesen (2015) suggest that when gatekeeping becomes ineffective, competition is intensified due to an increased number of game mod developers. Facing intensified competition, developers may find their efforts in offering mods less justifiable by the expected gains, and thus may reduce the frequency of submitting new mods or updates. In this study, we contribute to this stream of platform governance research by empirically examining how gatekeeping may shape an important type of complementor activity, knowledge sharing.

Knowledge sharing among complementors

One type of complementor activity worthy of attention in the context of platform ecosystems is knowledge sharing. Knowledge sharing among complementors is valuable to platforms because it helps complementors better understand user needs, which in turn can improve their odds of providing high-quality product offerings (Griffin & Hauser, 1992; Kapoor & Agarwal, 2017; Vasilescu, Filkov, & Serebrenik, 2013). Knowledge sharing is especially important for innovation-driven platforms. Innovation is a process of (re)combination of knowledge components, and knowledge sharing is a key link in the process (Fleming, 2001; Majchrzak *et al.*, 2004). For instance, for mobile app platforms such as Android or iOS, knowledge sharing among app developers about app development tools, programming languages, and bug fixing tips can lead to faster development cycles of high-quality apps (Pan, Kim, & Whitehead, 2009).

However, research on knowledge sharing among complementors remains scarce. One important reason might be that such activities mostly happen off the platform, in the form of offline meetups or information exchange on third-party forums. Hence, they are often beyond the control of the platform owner and hard to observe and track (Greenstein & Nagle, 2014). Moreover, knowledge sharing activities are usually qualitative in nature (in the form of unstructured texts), making themselves difficult to quantify and measure (Stavrianou, Andritsos, & Nicoloyannis, 2007). In this study, we address these challenges by tracing the "Internet trails" of knowledge sharing activities on two online forums – StackOverflow and Reddit – and by employing multiple methods (e.g., econometric analysis and text mining). Specifically, we contribute to platform ecosystems research by analyzing how individual-level knowledge sharing activities among app developers may change after an exogenous change in a platform's gatekeeping policy.

How platform gatekeeping affects complementors' knowledge sharing

We propose that gatekeeping could affect complementors' knowledge sharing activities through two mechanisms: market competition and community collaboration (Wareham *et al.*, 2014). First of all, as market competitors, complementors are mainly driven by extrinsic motivations such as maximizing monetary profit, signaling capability for career advancement and learning new practical skills for future value propositions (Davenport & Beck, 2001; Falkinger, 2008). Under this role, complementors consider knowledge to be a critical asset to accomplish valuable innovation goals (Fleming, 2001). Therefore, when competition intensity is high, complementors strive to gain a competitive advantage in innovation creation and maximize their own monetary or financial benefit. As a result, they are less likely to share knowledge.

The existing literature suggests that gatekeeping could affect competition intensity and knowledge sharing in two ways. For example, Boudreau and colleagues (2009; 2015) suggest that gatekeeping lowers the intensity of market competition by reducing the number of complementors. As competitions are eased, complementors may become less sensitive to knowledge protection and more motivated for knowledge sharing (Liebeskind *et al.*, 1996). Conversely, Zhu and Iansiti (2013) find that preventing more complementors from joining the platform reduces the variety and availability of complementary product offerings, thus losing consumers. With fewer consumers available, competition among complementors could be intensified, leading to less knowledge sharing. These mixed predictions suggest an important yet under-explored research gap regarding the interaction between knowledge sharing and market competition in the platform context.

At the same time, complementors could collaborate with each other to develop product offerings for platforms (Wareham *et al.*, 2014). As community collaborators, complementors are primarily driven by intrinsic motivations, such as community identity, social embodiment (Bagozzi & Dholakia, 2006; Beck, Pahlke, & Seebach, 2014; Faraj *et al.*, 2011; Wasko & Faraj, 2005), or professional pride bestowed by a strong community identity (Bagozzi & Dholakia, 2006; Shah, 2006), and hence are more likely to share knowledge.

By controlling who could access the platform, a gatekeeping policy can affect knowledge sharing by shaping the collaboration orientation of complementors. However, the existing virtual community literature could not make definitive directionality postulations for this relationship. For instance, Shah (2006) and Faraj and Johnson (2011) suggest that a larger number of community members might lead to an increased likelihood of reciprocity within a community, thereby increasing collaboration orientations. Hence, gatekeeping that reduces the number of complementors may hinder their knowledge sharing activities. In contrast, Bagozzi and Dholakia (2006) argue that a change of the member composition could hurt collaboration orientations by obscuring the identity of a virtual community and reducing members' perceived social obligation to contribute to the community. Therefore, gatekeeping, which helps maintain a stable or consistent community identity, could sustain or even enhance knowledge sharing activities. These contradictory predictions in the literature call for a carefully-designed empirical examination of how platform gatekeeping affects knowledge sharing, to be described in detail in the following section.

DATA AND METHODS

Complementors' off-platform knowledge sharing on online forums

Complementors' off-platform knowledge sharing is often difficult to observe and measure because they are hardly captured by platform-provided tools or information (e.g., page views, submission frequency, or product ranking). Fortunately, third-party online forums that complementors may visit or use can provide rich information about their knowledge sharing activity (Faraj *et al.*, 2011). This observation provides us with an opportunity to identify an empirical context for this study: Specifically, our empirical analysis leverages a carefully-created dataset tracking a sample of iOS and Android app developers (complementors of the iOS and Android platforms) and their knowledge sharing activity on two major online forums for IT professionals: StackOverflow.com and Reddit.com.

This empirical context is appropriate for us for several reasons. First, app developers are active users of online technical forums, which they frequently visit to share knowledge with peers, allowing us to capture their "Internet trails" for empirical analysis. In addition, our focus on app developers for iOS and Android reduces any selection bias, as iOS and Android are the dominant platforms for mobile devices. Second, for StackOverflow.com, web traffic records show that the site receives frequent contributions from app developers in major economies such as the U.S, Europe, Japan, and India (Vasilescu *et al.*, 2013). Moreover, developers of different types and diverse backgrounds, from fully-employed to indie developers, participate in the forum, easing concerns that certain types of programmers may self-select into this forum. Third, detailed user logs of StackOverflow.com provide rich data for analysis of the various steps involved in knowledge sharing (Nasehi *et al.*, 2012; Oktay, Taylor, & Jensen, 2010). Kuk (2006) finds that digital logs of Internet servers faithfully record user activities. Like other Internet servers, StackOverflow.com records detailed user activities, such as asking questions and posting answers, in virtually real time, allowing us to examine app developers' knowledge sharing in a fine-grained manner.

Despite many of the advantages it offers, we acknowledge that StackOverflow.com is not the only active forum that app developers visit for app development related discussions. Thus, in addition to using StackOverflow.com as our primary data source, we also use Reddit.com, another active online forum, as a secondary empirical context and source of data. Although Reddit.com does not provide information on the various steps of knowledge sharing, it hosts several active subforums ("subreddit" hereinafter) for fine-grained topics related to app development. One such topic, jailbreak, is closely related to our identification strategy discussed below. Thus, apart from allowing us to test the robustness of our results from the StackOverflow.com data, Reddit.com also proves to be valuable in helping us assess some key assumptions and robustness of this study, as we discuss in the Results section later.

Research design and identification strategy

To examine the effects of gatekeeping policy on complementors' knowledge sharing requires a research design that can effectively address endogeneity related concerns. Endogeneity may originate from several sources and weaken the study's effectiveness in identifying the causal effects. For example, certain programmers may self-select into, say, the iOS platform for unobserved reasons (Heckman, 1979), such that insightful questions and quality answers are more likely to be posted on StackOverflow.com or Reddit.com by iOS programmers. As another example, knowledge sharing among app developers may trigger a platform owner's decision to adopt particular gatekeeping policies to encourage or discourage particular developers from providing complementary offerings. To address these concerns, we adopt a quasi-experimental design in which we exploit the unexpected deficiency of Apple's gatekeeping policy, the "jailbreak" of iOS 7, as an exogenous shock to the app developers of the iOS platform (the treatment group) and compare their knowledge sharing activity with that of the shock-immune app developers of the Android platform (the control group), before and after the jailbreak, as illustrated in Figure 1. This difference-in-differences (DD) approach can help reduce potential endogeneity and strengthen the validity of the results (Angrist & Pischke, 2009; Oktay *et al.*, 2010).

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

A brief explanation about jailbreak and the context is in order. The jailbreak of the iOS operating system refers to the hacking process that exploits certain loopholes in iOS to remove some of Apple's built-in restrictions, allowing users of iOS devices to install apps from developers not recognized by Apple's App Store (and thus not officially approved by Apple). A very small group of elite hackers develop this hacking process and program it to be fully automated so that laymen without sophisticated computer skills can also easily jailbreak their iOS devices. When an automated jailbreak program is widely available,[1] average jailbreak app developers without advanced programming or hacking skills may effectively join the iOS app developers' community and bypass Apple's gatekeeping policy to publish their apps designed for users of jailbroken devices (Zdziarski, 2012).[2] However, the amount of time required to find ways to jailbreak iOS can be unpredictable because, for each newer version of iOS, Apple adds additional protection from jailbreak.

The jailbreak of iOS 7 fits our research purpose particularly well for several reasons. First, unlike the case in prior versions of iOS, the timespan between the official release of iOS 7 and the release of its jailbreak program was exceptionally long (95 days, see Table 1 for details), pointing to the high levels of technical difficulty and uncertainty of achieving this jailbreak. In addition, we search and examine any technical discussions about perceived uncertainties about a successful jailbreak at the time. As suggested by an online post (Anthony, 2013), there was a generally pessimistic sentiment about the immediate availability of jailbreak of iOS 7 at the time. Thus, it is reasonable to consider that Apple's gatekeeping policy is unexpectedly weakened by the jailbreak in December 2013. Second, the exceptionally long timespan (95 days) allows us to build a sufficiently long panel dataset tracking app developers' knowledge sharing activity before the treatment (before the release date of the jailbreak program, yet after the official release date of iOS 7) and after the treatment, for empirical analysis.

-----Insert Table 1 about here-----

In our design, app developers for Android serve as a control to the treatment group (i.e., iOS app developers), given that Android app developers are not affected by the jailbreak. Unlike iOS, Android's open platform policy contains very few quality standards for publishing apps (Tiwana, 2013).

[1] Once a jailbreak program is developed, it is publicly and freely available online. Jailbreak is a significant event for the iOS platform, and thus receives extensive coverage by major IT news media. As a result, the time lag between the release of a jailbreak program and it being widely available is very short and hence negligible in our research context. For details, please see: http://www.forbes.com/sites/anthonykosner/2013/02/10/what-7-million-jailbreaks-are-saying-is-apple-listening

[2] In theory, it is also likely that an existing iOS app developer may join jailbreak app developers and leave the iOS developers' community. However, in reality, because Apple strictly forbids such "switching" by revoking the developer's development authentication once identified, few iOS app developers will want to switch to develop for jailbroken devices.

Other than the differences in platform gatekeeping policies, prior research suggests that app developers for these two platforms are comparable in general (Kapoor & Agarwal, 2017). Moreover, a focus on developers who choose to engage in knowledge sharing online on StackOverflow.com or Reddit.com further enhances comparability between the treatment group and control group because of similarities in general interaction logics on online forums and forum-wide etiquettes. In addition, within our study's time window, we find no similar platform-wide shock happening to the Android platform.[3] Thus, by using a difference-in-differences approach, we can isolate the causal effects of the exogenous deficiency in gatekeeping, as identified by the jailbreak of iOS 7, on changes in iOS app developers' knowledge sharing activity, compared with those of Android app developers.

Note that in addition to constructing the main treatment group and control group above, later we also implement Coarsened Exact Matching (CEM) to create a matched sample to further strengthen the assumptions of a quasi-experimental design. Prior research suggests that CEM offers advantages over alternative matching methods because of its incorporation of monotonic balance bounding (Blackwell *et al.*, 2009) that reduces model-dependence, bias, and inefficiency (Iacus, King, & Porro, 2012), and is increasingly used in management research in recent years (Younge, Tong, & Fleming, 2015).

Sample construction

StackOverflow.com sample. The primary data source is obtained from an online data dump of StackOverflow.com. Although this specific data source is rarely used in the strategic management literature, similar data sources have been used in prior management research. For instance, Kuk (2006)

[3] In the Android ecosystem, there is a concept called "rooting", which may sometimes be confused with, but is distinct from, jailbreak. Jailbreak allows a user to use software that Apple doesn't authorize on iOS-based devices (e.g., iPhones). Rooting, on the other hand, gives a user access to more or less the entire operating system. For instance, a user can remove the whole operating system and replace it with a user-created operating system that contains tweaks and enhancements. Apart from such technical differences, the magnitude of the impact also differs between the two. A jailbreak event often has an ecosystem-wide impact: once a jailbreak is achieved, the majority of the devices running the affected iOS versions may be jailbroken. However, rooting is often manufacturer- or device-specific. For example, a Samsung device may need specific software for rooting processes, while an LG device does not need third party software to root. For these reasons, rooting in the Android ecosystem will have little impact on any ecosystem-wide change in app developers' activities.

uses similar digital records to study user activities online. Studies using server logs in various research contexts have also shown the validity of this type of data source (Sun & Zhu, 2013; von Krogh & von Hippel, 2006). To construct the base sample, we start with all StackOverflow.com users who have posted at least one question with 500 or more views on the iOS or Android subforum within 365 days before September 18, 2013 (the start date of our sample's time window, see below). The 2013 website statistics of StackOverflow.com suggest the presence of many old, inactive users who rarely post questions, and including such users in our sample can introduce noise and bias; as a result, we limit the sample to recently active users. Moreover, we include a required lower limit of view count to exclude users who post trivial questions that are rarely viewed.[4] Then, we collect Internet log data for the above app developers for a total of 27 weeks (13 weeks before and 13 weeks after the treatment, as well as the treatment week) from September 18, 2013, to March 26, 2014, leading to a sufficiently long panel dataset surrounding the jailbreak event. Since the jailbreak of iOS 7 happened on December 21, 2013, and a typical app developing cycle ranges from 8 to 12 weeks,[5] the length of the time window is reasonable for covering a full developing cycle for apps. The data harvesting process was performed four weeks after the 27th week of the time window because although app developers' questions are posted in real time, more time is needed for the questions to receive attention from other developers on StackOverflow.com.

One potential issue is related to multi-homing app developers. Theoretically, developers can create apps for both the iOS and Android platforms. However, although the general programming logic is similar for iOS and Android, differences in *software development kits (*SDKs), application programming interfaces (APIs), and programming syntaxes between the two make such multi-homing developers rare. Still, to address potential effects of multi-homing developers, we exclude all app

[4] The average view count for all questions between 2008 and 2012 in the iOS or Android sub-community is about 1,300, with a standard deviation of about 4,000. Thus, a requirement of 500 or more views will introduce little bias in the dataset.

[5] For more details, see sources included in this link: http://www.accella.net/iphone-app-development-timeline.

developers who post questions on both the iOS and Android subforums, removing 55 multi-homing developers from the sample.

We also retain only app developers within the top 20 percent activity level online (based on the distribution of question-posting frequency in the study's time window) to achieve a broad coverage of active developers and to exclude developers who do not consider this forum as a main channel of knowledge sharing, because prior research suggests that the majority of the activity in online communities comes from users within the top 10 percent activity level and occasional users might introduce noise to the records of their activities (Kuk, 2006). Our final main sample, after these steps, consists of 418 iOS app developers and 841 Android app developers.

Reddit.com sample. The secondary data source was obtained from the posting history of Reddit.com stored on Google Bigquery (https://cloud.google.com/bigquery) (Reddit, 2016). Because the Reddit.com data do not come with post view counts, we start with all Reddit.com users who posted at least once, within 365 days before September 18, 2013, in the "subreddits" related to iOS and Android app development as well as jailbreak development. We perform two analyses using the Reddit.com data within the same 27 weeks of time window as the StackOverflow.com sample. First, we use the posting activities of users in subreddits related to jailbreak development to empirically assess our assumption that there is an influx of jailbreak app developers that changes the composition of the iOS developers' community. Unlike StackOverflow.com where jailbreak-related discussions are extremely rare, Reddit.com hosts active technical discussions regarding jailbreak related app development, allowing us to conduct the analysis.

Second, we use the posting activities of users in subreddits related to iOS and Android app development as a second data source (in addition to StackOverflow.com) to examine the robustness of our findings. Because a post on Reddit.com can be an original post, a reply to the original post, or a reply to a reply, it is not possible to clearly differentiate knowledge soliciting from knowledge providing; as a result, we can only use the Reddit.com data to triangulate our main analysis at the aggregated post level (i.e., *Post Count* and *Post Quality*), but not the disaggregated question and answer levels. Similar to the construction of the StackOverflow.com sample above, we retain only those developers within the top 20 percent activity level on Reddit.com.

Variables and measurement

Unit of analysis. Following prior studies using data sources of a similar nature (Kuk, 2006; Sun & Zhu, 2013), the unit of analysis in this study is individual app developer by week. Specifically, we prepare a panel-structured data set that records the weekly knowledge sharing activities of app developers on the iOS and Android subforums of StackOverflow.com and Reddit.com.

Dependent variables. We use the user activity data of the iOS and Android subforums from StackOverflow.com and Reddit.com to construct our dependent variables and examine the treatment effect of jailbreak on the aggregated level of knowledge sharing by app developers. Knowledge sharing can be conceptualized and measured in terms of quantity and quality. For instance, soliciting knowledge can be represented by the number of questions asked and the quality of those questions. Our data sources afford a unique opportunity to examine both the quantitative and qualitative aspects of knowledge sharing, as shown by the two sets of dependent variables below.

The first set of dependent variables utilizes data from both StackOverflow.com and Reddit.com to measure the aggregated level of knowledge sharing. The first dependent variable, *Post Count*, measures the quantitative aspect of knowledge sharing. For StackOverflow.com, we measure this variable by counting the total number of questions and answers posted by an app developer on the iOS or Android subforum in a given week; for Reddit.com, this is measured by the total number of posts posted by an app developer on the iOS or Android related subreddit. The second dependent variable, *Post Quality*, measures the qualitative aspect of knowledge sharing. We measure the variable by the average score of the posts as defined above by an app developer on StackOverflow.com or Reddit.com. Both StackOverflow.com and Reddit.com employ a scoring system that allows other users to collectively determine the quality of a given post, question, or answer by voting. When a user finds a given post insightful or important, she may choose to “upvote” for the post, increasing the total score the post receives; she may also choose to “downvote” for the post if she finds it unhelpful or potentially misleading. Thus, higher-quality posts tend to receive higher scores.

The second set of dependent variables utilizes StackOverflow.com data only, to measure various steps involved in the process of knowledge sharing. The first dependent variable, *Question Count*, measures the quantity aspect of soliciting knowledge. We measure this variable by counting the total number of iOS/Android questions posted by an app developer in a given week. The second dependent variable, *Question Quality*, measures the qualitative aspect of soliciting knowledge. We measure this variable by the average score received for the iOS/Android questions posted by an app developer in a given week. The third dependent variable, *Answer Count*, measures the quantity aspect of providing knowledge. We measure this variable by the total number of answers to the iOS/Android related questions posted by an app developer in a given week. The fourth dependent variable, *Answer Quality*, measures the quality aspect of providing knowledge. We measure this variable by the average score received for the answers to the iOS/Android questions posted by an app developer in a given week. As with the question voting system explained earlier, StackOverflow.com also allows users to vote for answers that they find insightful or useful. The fifth dependent variable here, *Accepted Answer Quality*, measures the overall outcome of knowledge sharing. We measure this variable by the average score of the accepted answers among those answers to iOS/Android questions posted by an app developer in a given week. Although StackOverflow.com allows multiple answers to be posted for a given question, each question can only have one accepted answer. This answer is either adopted by the original app developer posting the question or is recommended by StackOverflow.com based on the scores received to indicate the usefulness of the answer in addressing the focal question.[6] Thus, the

[6] We inspect users’ comments for 50 randomly-selected accepted answers from the sample, and find that all accepted answers

quality of the accepted answer may represent the overall quality of the outcome of knowledge sharing.

Explanatory variables. Following the quasi-experimental design, the DD "treatment group" variable, *iOS*, is a dummy variable that takes a value of 1 for iOS app developers, and 0 for Android app developers. The DD "after" variable, *After*, is a dummy variable that takes a value of 1 for weeks after the jailbreak event, and 0 for weeks before the event (including the 'focal' week of jailbreak). Then, the interaction term *DD* (*iOS*After)* is created to identify the potential treatment effect of deficient gatekeeping (Angrist & Pischke, 2009; Meyer, 1995; Sun & Zhu, 2013).

Other covariates. For the sample drawn from the StackOverflow.com data, we include several covariates to further control for any differences between the treatment group and control group. Specifically, we include the average count of question views (*Question View Count*) and average count of question comments (*Question Comment Count*) for all questions posted by an app developer in a given week, because these question-related variables may also explain the variance of the dependent variables (Sun & Zhu, 2013). Although we allow four extra weeks (following the 27-week time window) for all questions to receive enough attention, including additional controls related to the questions posted can further isolate the causal effect in question. In addition, prior research has suggested that knowledge sharing activities may correlate with other social activities on an online community (Faraj & Johnson, 2011). Thus, social interactions in the form of editing posts, commenting on others' posts, bookmarking posts, or replying posts may affect an app developer's knowledge sharing activity. We therefore include app developers' editing frequency (*Edit Activity*), commenting frequency (*Comment Activity*), favorite-adding frequency (*Add-Favorite Activity*), and overall answering frequency (*Answer Activity*), to control for these time-varying factors related to an app developer, because these factors can affect users' effort in contributing to online communities (Roberts, Hann, & Slaughter, 2006). For the sample drawn from the Reddit.com data, we are not able to calculate

receive at least one comment indicating that the solutions provided effectively address the focal question.

and include these covariates, however, because no time-varying information can be extracted from this data source.

For both the StackOverflow.com sample and Reddit.com sample, we incorporate a full set of individual app developer fixed effects to remove effects due to any unobserved, time-invariant characteristics of app developers (e.g., individual intelligence level or background) (Wooldridge, 2010). We also include a full set of week fixed effects to control for any environmental factors or week-by-week variations that may affect our results. In all regressions, robust standard errors are clustered at the individual app developer level (Wooldridge, 2010). Table 2 provides a list of all variables and their definitions and descriptive statistics.

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

Model specification

Given the difference-in-differences research design, we estimate the following model specification to examine the treatment effect of a deficiency in gatekeeping on knowledge sharing among app developers:

$$Y_{it} = \beta_0 + \beta_1 DD_{it} + \delta X_{it} + T_t + I_j + \varepsilon_{it} \quad (1)$$

where Y_{it} is the relevant dependent variable for app developer *i* in week *t*; β_0 is the intercept; β_1 identifies the treatment effect of DD_{it} (*iOS*After*); X_{it} are covariate controls; T_t is a vector of week fixed effects; I_j is a vector of individual app developer fixed effects; and ε_{it} is the error term. We run linear models for all dependent variables: *Post Count*, *Post Quality*, *Question Count*, *Question Quality*, *Answer Count*, *Answer Quality*, and *Accepted Answer Quality*. Among them, *Post Count*, *Question Count*, and *Answer Count* are count variables, while others are continuous variables; when we run count models (negative binomial models) for the two dependent variables, we obtain qualitatively similar results.

RESULTS

Assessment of assumptions

One assumption underlying our exploration is that an unexpected deficiency in platform gatekeeping, via its impact on both the size and composition of complementors, may trigger changes in complementors' knowledge sharing activity. In our empirical context, such changes may be due to a possible influx of jailbreak app developers into the iOS developers' community after the jailbreak of iOS 7. We examine this assumption both quantitatively and qualitatively. Quantitatively, we examine the extent to which jailbreak app developers join online discussions of iOS-related subreddits after the jailbreak. We find that, within the 13-week time window after the jailbreak, 130 Reddit.com users who posted jailbreak related posts before (out of 1,795 active users on iOS-related subreddits) posted 1,820 (out of 25,130) posts on iOS-related subreddits. This finding indicates that some app developers who were interested in jailbreak related issues indeed have shifted their development focus and joined the iOS developers' community after the jailbreak.

Qualitatively, we perform text analysis of posts of Reddit.com users on jailbreak related subreddits, by using the LDA model (Blei et al., 2003). LDA is one of the most popular topic modeling approaches for extracting latent topics and their varying presences in each document from a large corpus. LDA assumes that each document is generated from a mixture of topics, with each topic corresponding to a multinomial distribution over all the words in the vocabulary. Although its applications in management research remain scant, LDA has been used widely in other fields for mining topics in different types of corpora, including articles (Blei & Lafferty, 2007) and news (Yang, Torget, & Mihalcea, 2011). In our study, we use LDA to uncover the topics, represented by a mixture of words, discussed by jailbreak related posts on Reddit.com before and after jailbreak. As a standard procedure, we preprocess the data following the best practices in LDA modeling (Blei *et al.*, 2003) by removing non-English words and special characters (e.g., URLs, punctuation), filtering out "stop words” (e.g., “an,” “the,” “these”) as well as terms that occur in less than 2% of the posts in our data. We stem each word to its root using Porter Stemmer (Porter, 1980). After several empirical tests in an iterative process and following the suggestion of Blei et al. (2003), we set the number of topics to be 50. As a result, we build a topic model with 50 topics using all the relevant posts on Reddit.com pre- and post-jailbreak. Each topic is represented by a list of words with their presence probability. We then follow prior research (Yang & Pedersen, 1997) to use “information gain” to select the words that have significant changes before and after jailbreak and use them to construct a word cloud (see Figure 2). As shown in the figure, there is a well-noticeable increase in the frequency of topics/keywords associated with iOS related app development among the jailbreak app developers. Specifically, before the jailbreak of iOS 7, jailbreak app developers are more likely to discuss technical topics related to jailbreak process or tweak development. After the jailbreak, the topics discussed cover more iOS or iPhone related development. This qualitative change suggests that their attention shifts from technical challenges of jailbreak to the actual development of jailbreak tweaks as apps for the iOS platform.

-----Insert Figure 2 about here-----

Another key assumption of our research design requires that iOS and Android app developers are comparable. We assess this assumption in several ways. First, although the syntaxes for creating apps vary by platform, app developers should possess some general knowledge of app development. We examine this idea by comparing the tags most frequently used by iOS and Android app developers on StackOverflow.com. As shown in Panel A of Table 3, iOS and Android app developers share some general programming knowledge common to IT professionals, as indicated by such tags as object-oriented programming (e.g., Java), web application (e.g., Ruby-on-Rails), and Structured Query Language (e.g., MySQL). Meanwhile, tags unique to the iOS developers or Android developers are often about platform-specific syntaxes. For instance, “ui-view-controller” and “android-intent” are two platform-specific topics for the iOS and Android platforms, respectively.

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

Second, when the gatekeeping policy is made deficient by jailbreak, app developers should exhibit comparable patterns of knowledge sharing. Specifically, we compare the means of the dependent variables for the iOS and Android app developers *before* and *after* the jailbreak of iOS 7. As shown in Panel B of Table 3, before jailbreak, there are noticeable differences between iOS and Android app developers in their knowledge sharing activities. After jailbreak, such differences disappear, suggesting that iOS and Android app developers indeed show similar patterns of knowledge sharing. As expected, the univariate DD analysis in the last column points to a significant drop in the values of the dependent variables, providing initial evidence of a deleterious effect of deficient gatekeeping policy on complementors' knowledge sharing activities.

Third, we examine covariate balance between iOS and Android app developers. As shown in Panel C of Table 3, the differences in the values of the covariates are nonsignificant, indicating that the two groups of app developers in our sample are comparable to each other. Finally, although not reported in tables, we also compare the two groups of app developers in terms of their forum reputations, post view counts, and total count of upvotes and downvotes received within one year before the start of our study's time window. Differences between the two groups are again nonsignificant (i.e., -1.13, 0.32, and 0.57, respectively), indicating that these app developers are comparable in their historical activities and characteristics on StackOverflow.com.

Main results

We now move on to report main results from the regression analysis. Table 4 reports DD regression results focusing on aggregated knowledge sharing using both the StackOverflow.com sample and Reddit.com sample. Columns 1 and 2 use the StackOverflow.com sample, and Columns 3 and 4 use the Reddit.com sample. We use models in Columns 1 and 3 to explore the quantitative aspect of the relation between a deficient gatekeeping policy and knowledge sharing among complementors, and results in Columns 2 and 4 to explore the qualitative aspect.

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

The results in Columns 1 and 3 suggest that when the gatekeeping policy is weakened by the jailbreak of iOS 7 and jailbreak app developers join the iOS app developers' community, existing iOS app developers engage in knowledge sharing less frequently. Specifically, the negative coefficient on the DD variable in Columns 1 indicates that iOS app developers post 0.16 fewer posts, about eight percent drop in terms of economic magnitude, on StackOverflow.com after the jailbreak event, compared with Android app developers. The associated p-value of 0.00078 suggests that the probability of observing a drop in post frequency at least as extreme as the one in our empirical result, assuming the truth that a deficiency in gatekeeping triggers no change in complementors' knowledge sharing frequency, is very low. Similarly, the negative coefficient of -0.04 on the DD variable in Columns 3 and its associated p-value of 0.054 suggest a drop in knowledge sharing frequency on Reddit.com, an observation that is also unlikely due to random sampling errors.

The results in Columns 2 and 4 suggest that when the gatekeeping policy is weakened by the jailbreak of iOS 7, the quality of knowledge sharing among existing iOS app developers decreases. Specifically, the negative coefficient on the DD variable in Columns 2 indicates that the ratings of iOS app developers' posts reduce by 0.07 units, about 16 percent drop in terms of economic magnitude, on StackOverflow.com after the jailbreak event, compared with Android app developers. The associated p-value of 0.00084 suggests that the probability of observing a drop in post quality at least as extreme as the one in our empirical result, assuming a null effect that a deficiency in gatekeeping triggers no complementors' knowledge sharing quality, is very low. Similarly, the negative coefficients of -0.04 on the DD variable in Columns 4 and its associated p-value of 0.037 suggest a drop in knowledge sharing quality on Reddit.com, an observation that is also unlikely due to random sampling errors.

We seek to further unravel the process of knowledge sharing by leveraging the rich information on the various steps involved in knowledge sharing as recorded by the "Internet trails" on StackOverflow.com. Specifically, we examine how the jailbreak affects three steps of knowledge sharing: knowledge solicitation (measured by *Question Count* and *Question Quality*), knowledge provision (measured by *Answer Count* and *Answer Quality*), and the outcome of knowledge sharing (measured by *Accepted Answer Quality*). Based on the above empirical findings, we suspect that when the gatekeeping policy is weakened by the jailbreak of iOS 7 and jailbreak app developers join the iOS app developers' community, existing iOS app developers will raise fewer and lower-quality questions, and provide fewer and lower-quality answers to questions of other iOS app developers; as a result, the quality of the accepted answer will also become lower. Table 5 reports DD regression results for the five finer-grained dependent variables using the StackOverflow.com sample. The results are consistent with our findings reported in Table 4, as shown by the negative coefficients on the DD variable in Columns 1-5. Specifically, compared to Android app developers, iOS app developers raise fewer questions and the quality of these questions also drops, after the jailbreak event. Meanwhile, these developers offer fewer and lower-quality answers to fellow developers. Finally, the quality of the accepted answers from these developers also declines. The associated p-values, 0.035, 0.0003, 0.024, 0.002, and 0.0002 respectively, suggest that observing effects at least as extreme as the above is unlikely due to chance. Interpretations of the estimated coefficients indicate that after jailbreak, iOS app developers on average ask four percent fewer questions, and the quality of these questions drops by 26 percent; these developers also provide 10 percent fewer answers, the quality of these answers drops by 16 percent, and the quality of accepted answer drops by 24 percent. These observed changes are considered quite substantial, given that Apple's gatekeeping policy is weakened but not completely voided by the jailbreak of iOS 7.

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

Robustness checks

We examine the robustness of our results in several ways. First, we implement CEM to create a matched sample using the StackOverflow.com data. Specifically, we match on all covariate controls as differences in the covariates may affect developers' knowledge sharing activity. We then fit the models in Tables 4 and 5 to the matched sample. As Table 6 shows, we continue to find coefficients on the DD variable to have signs, magnitudes, and levels of p-values that are consistent to the main analysis, indicating that our results are robust to both unmatched and matched samples.

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

Second, we test the robustness of our results using app developers of alternative activity levels. As suggested by Kuk (2006), the majority of contributions in online communities come from users within the top 10 percent activity level. In the main analysis, we focus on app developers within the top 20 percent activity level for a broad coverage of developers. In additional analysis, we create three new samples by focusing on app developers of three other activity levels (5%, 10%, and 15%, respectively) using the StackOverflow.com data, and fit the models in Tables 4 and 5 to those samples. Table 7 reports summary results for the DD variable to conserve space; for ease comparison, the bottom row reproduces results from the main analysis (i.e., top 20 percent activity level). As the table shows, the coefficients on the DD variable continue to have signs, magnitudes, and levels of p-values that are qualitatively consistent to the main analysis, indicating that our results are robust to app developers of different activity levels.

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

Third, we test the robustness of our results using a different control group than Android app developers. Because of strong competition between the iOS and Android ecosystems, a major shock to the iOS ecosystem may have indirectly affected the dynamics on the Android platform and the activities of its complementors. To address this concern, we create another sample consisting of Windows Phone OS app developers, which serve as an alternative control group. Applying the same model specifications in Tables 4 and 5 and using this new control sample, we continue to find that the statistics for the DD variables in Table 8 are consistent with the main analysis.

-----Insert Table 8 about here-----

DISCUSSION

Strategic management research on platform governance and its implications for complementors remains scarce (McIntyre & Srinivasan, 2017). In this study, we extended extant research by studying the relationship between platform gatekeeping, a prominent governance policy, and complementors' knowledge sharing activity (Adner, 2017; Cennamo & Santalo, 2013; Tiwana, 2013). Given that the existing literature provides no clear guidance about the directionality of this relationship, we were determined to examine this relationship empirically by leveraging a natural experiment—an exogenous change of a platform's gatekeeping policy in the case of the jailbreak of Apple's iOS 7. We traced the knowledge sharing activities of a sample of active iOS and Android app developers on two online forums, StackOverflow.com and Reddit.com. Employing a difference-in-differences approach and text-mining techniques, we showed strong evidence that an exogenously deficient gatekeeping policy of Apple – caused by the Jailbreak of iOS – reduced the quantity and quality of knowledge sharing among iOS app developers.

Potential mechanisms

Based on the empirical results and existing literature, we suggest two potential mechanisms, market competition and community collaboration, to explain the documented relationship between platform gatekeeping and complementors' knowledge sharing. From a market competition perspective, our findings corroborate the mechanism proposed by Boudreau and colleagues (2009; 2015). The sudden emergence of jailbreak developers may intensify the competitions on the app development platform. For example, some apps of iOS developers could be quickly replicated by jailbreak developers with strong execution abilities, who could publish those apps at a lower price or even for free (Li, Singh, & Wang, 2014). Those similar yet cheaper jailbreaking apps might attract more users, thereby posing a direct threat to the monetary benefits of existing iOS developers. Also, prior studies have found that a small group of jailbreak developers are highly skillful at implementing complex and novel software applications (Zdziarski, 2012). Allowing those developers to enter the previously gated platform and bring in more novel apps would further intensify the competition. The intensified competition dynamics would stimulate iOS developers' extrinsic motivations to maximize their own monetary benefit; as a result, they would devote less effort to sharing knowledge on the forums.

We also attempt to explain why some alternative market competition mechanisms do not support our empirical findings. According to Zhu and Iansiti's (2013) arguments, a deficient gatekeeping may result in an increase in the quantity and variety of product offerings (supply) on the platform, which can trigger an indirect network effect to attract more consumers (demand), lessening the sense of competition among complementors. We believe this mechanism may be less applicable in our context since the number of iOS consumers is less likely to increase significantly within such a short period after the jailbreak. In this sense, our study has furthered the existing understanding of the market competition mechanism in the platform governance context: if the growth rate of consumers triggered by the new product offerings by new complementors could not satisfy the demand of existing complementors, competitions among complementors would be intensified, and knowledge sharing activities would likely decrease in quantity and quality.

From a community collaboration perspective, we believe that the mechanisms suggested by Kuk (2006) and Bagozzi and Dholakia (2006) are more applicable. An influx of jailbreak developers would increase the heterogeneity of the app developer community, which, previously, was mainly made up by iOS developers. The increased heterogeneity may obscure the community values and identities. For instance, Jailbreak developers are often known as hackers who appreciate technological "freedom" and "open source" development (Coleman, 2013; Cross, 2006), which may contradict the community values of iOS developers, as they tend to follow Apple's developer protocols and would like to protect their proprietary apps and sustain their market (Apple, 2014). Such increased heterogeneity among community members could hinder the effectiveness and efficiency of the knowledge sharing process (Kuk, 2006), as well as dampen iOS developers' intrinsic motivations such as community value, community-related prestige, or social obligation to collaborate with each other

(Bagozzi & Dholakia, 2006; Beck *et al.*, 2014), both of which lead to a decrease in the quantity and quality of knowledge sharing among iOS developers.

Alternatively, the reciprocity mechanism suggested by Shah (2006) and Faraj & Johnson (2011) is less applicable in this context. This stream of research implies that a large number of participants could generate a higher level of overall activity, thereby increasing the level of direct or indirect reciprocity among members. However, the big differences between jailbreak developers and iOS developers (e.g., culture, motivations, and values) would impede the establishment of such reciprocity. Consequently, we propose that the applicability of different community collaboration mechanisms in the context of gatekeeping and knowledge sharing may hinge upon the following boundary conditions: the similarity between incoming and existing complementors, and the heterogeneity (or homogeneity) of the complementor community.

Theoretical implications

Our study offers substantial theoretical implications. First, we contribute to platform governance research by showing that gatekeeping is an important governance policy that can significantly shape complementors' knowledge sharing activity. In addition, we discuss two potential mechanisms as well as their boundary conditions in explaining our empirical findings, which can enrich the theoretical link between platform governance and complementors' knowledge sharing. Second, we contribute to the platform network effects literature by showing that giving unconstrained platform access to complementors may not necessarily generate a competitive advantage but instead could negatively affect some critical activities of complementors such as knowledge sharing. We suggest that the changing dynamics of competition and collaboration among complementors be considered carefully when implementing a complementor growth strategy. Third, we contribute to research on platform complementors and their value creation activities by examining knowledge sharing, a hard-to-measure off-platform activity that is especially critical to innovation platforms. Given that complementors' on-platform and off-platform activities are often interrelated to each other, our study provides a first step toward a greater understanding of such relationships in future scholarship.

Empirical and practical implications

Our study makes several empirical contributions. First, we make use of two distinct data sources, StackOverflow.com and Reddit.com, to establish and triangulate empirical evidence. These two well-curated and fine-grained data sources cover a wide spectrum of topics, such as digital technologies, open-source programming, and gaming experience exchange. Hence, a joint use of these data sources can improve the empirical rigor of future work on strategic management issues in the context of the digital economy. Second, this study introduces a text mining technique, LDA, into the management literature. This technique is generally applicable to quantitatively obtaining latent semantic information from unstructured text data. The technique proves especially valuable in the current digitized world where unstructured text is widely available but remains underutilized due to the difficulty of information extraction.

Our research also has substantial practical implications. We provide insights for practitioners about the potential downside of (over)growing the complementors of a platform and the value of deploying gatekeeping policies to shape complementors' behaviors and value creation activities. These insights are particularly important for the owners of innovation platforms where knowledge sharing plays an important role. Our finding also affirms the value of the current practice of the iOS platform, indicating that Apple would benefit from continuing to enforce a strict app review process, to better develop its complementors and induce greater, higher-quality knowledge sharing activities among iOS developers.

Limitations and future directions

This study has several limitations that provide additional opportunities for future research. First, while we find that an exogenous deficiency of gatekeeping could impair complementors' knowledge sharing activities through market competition and community collaboration mechanisms, the empirical data are limited in our ability to tease out which of the two mechanisms may contribute more to the documented relationship. Related to this, a potential future direction is to study the relative magnitude of extrinsic motivations versus intrinsic motivations of complementors under the market competition and community collaboration conditions. Second, although the literature suggests that (off-platform) knowledge sharing is positively related to the effectiveness of the platform, our data do not allow a direct test treating platform effectiveness as a dependent variable. Hence, a natural extension of this study is to link complementors' off-platform knowledge sharing with their on-platform effectiveness or the attractiveness of the platform as a whole. Third, while we recognize that individuals tend to be different from one another and we explicitly controlled for this consideration by incorporating a full set of individual fixed effects thus removing time-invariant individual differences, this study is not able to identify the effects of specific time-varying differences among app developers. Thus, variances in such factors could be another interesting boundary condition for the documented effect of deficient gatekeeping on knowledge sharing patterns in our study. As platform ecosystems become increasingly prominent in the digital economy today, research on the relationship between platform governance and complementors' value creation activities will take on greater importance.

REFERENCES

Adner R. 2017. Ecosystem as Structure: An Actionable Construct for Strategy. Journal of Management 43(1): 39-58.

Angrist JD, Pischke J. 2009. Mostly Harmless Econometrics: An Empiricist's Companion. Princeton University Press: Princeton.

Anthony S. 2013. iOS 7 jailbreak for iPhone and iPad: When, if ever, will the jailbreak be ready?

Apple. 2014. App Programming Guide for iOS. https://developer.apple.com/library/ios/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/iPhoneAppProgrammingGuide.pdf.

Bagozzi RP, Dholakia UM. 2006. Open source software user communities: A study of participation in Linux user groups. Management Science 52(7): 1099-1115.

Baldwin CY, Clark KB. 2000. Design rules: The power of modularity. MIT press: Cambridge.

Beck R, Pahlke I, Seebach C. 2014. Knowledge Exchange and Symbolic Action in Social Media-Enabled Electronic Networks of Practice: A Multilevel Perspective on Knowledge Seekers and Contributors. Mis Quarterly 38(4): 1245-1270.

Blackwell M, Iacus S, King G, Porro G. 2009. cem: Coarsened exact matching in Stata. Stata Journal 9(4): 524-546.

Blei DM, Lafferty JD. 2007. A Correlated Topic Model of Science. Annals of Applied Statistics 1(2): 634-634.

Blei DM, Ng AY, Jordan MI. 2003. Latent Dirichlet allocation. Journal of Machine Learning Research 3(4-5): 993-1022.

Boudreau KJ, Hagiu A. 2009. Platform rules: Multi-sided platforms as regulators. Platforms, markets and innovation 1: 163-191.

Boudreau KJ, Jeppesen LB. 2015. Unpaid crowd complementors: The platform network effect mirage. Strategic Management Journal 36(12): 1761-1777.

Casadesus-Masanell R, Yoffie DB. 2007. Wintel: Cooperation and conflict. Management Science 53(4): 584-598.

Cennamo C, Santalo J. 2013. Platform competition: Strategic trade-offs in platform markets. Strategic Management Journal 34(11): 1331-1350.

Coleman EG. 2013. Coding freedom: The ethics and aesthetics of hacking. Princeton University Press: Princeton.

Cross T. 2006. Academic freedom and the hacker ethic. Communications of the Acm 49(6): 37-40.

Davenport T, Beck J. 2001. The attention economy: Understanding the new economy of business, Cambridge, MA: Harvard Business School Press.

Eisenmann T, Parker G, Van Alstyne MW. 2006. Strategies for two-sided markets. Harvard Business Review 84(10): 92.

Falkinger J. 2008. Limited attention as a scarce resource in information-rich economies. Economic Journal 118(532): 1596-1620.

Faraj S, Jarvenpaa SL, Majchrzak A. 2011. Knowledge Collaboration in Online Communities. Organization Science 22(5): 1224-1239.

Faraj S, Johnson SL. 2011. Network Exchange Patterns in Online Communities. Organization Science 22(6): 1464-1480.

Fleming L. 2001. Recombinant uncertainty in technological search. Management Science 47(1): 117-132.

Grant RM. 1996. Toward a knowledge-based theory of the firm. Strategic Management Journal 17: 109-122.

Greenstein S, Nagle F. 2014. Digital dark matter and the economic contribution of Apache. Research Policy 43(4): 623-631.

Griffin A, Hauser JR. 1992. Patterns of Communication among Marketing, Engineering and Manufacturing - a Comparison between 2 New Product Teams. Management Science 38(3): 360-373.
Gronli T-M, Hansen J, Ghinea G, Younas M. 2014. Mobile application platform heterogeneity: Android vs Windows Phone vs iOS vs Firefox OS. In Proceedings of the Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on.
Heckman JJ. 1979. Sample Selection Bias as a Specification Error. Econometrica 47(1): 153-161.
Iacus SM, King G, Porro G. 2012. Causal Inference without Balance Checking: Coarsened Exact Matching. Political Analysis 20(1): 1-24.
Kapoor R, Agarwal S. 2017. Sustaining Superior Performance in Business Ecosystems: Evidence from Application Software Developers in the iOS and Android Smartphone Ecosystems. Organization Science 28(3): 531-551.
Krugman PR. 1993. Geography and trade. MIT Press: Cambridge.
Kuk G. 2006. Strategic interaction and knowledge sharing in the KDE developer mailing list. Management Science 52(7): 1031-1042.
Li B, Singh P, Wang Q. 2014. Zoom in iOS Clones: Examining the Antecedents and Consequences of Mobile App Copycats. In Proceedings of the International conference on information systems.
Liebeskind JP, Oliver AL, Zucker L, Brewer M. 1996. Social networks, learning, and flexibility: Sourcing scientific knowledge in new biotechnology firms. Organization Science 7(4): 428-443.
Majchrzak A, Cooper LP, Neece OE. 2004. Knowledge reuse for innovation. Management Science 50(2): 174-188.
McIntyre DP, Srinivasan A. 2017. Networks, Platforms, and Strategy: Emerging Views and Next Steps. Strategic Management Journal 38(1): 141-160.
Meyer BD. 1995. Natural and Quasi-Experiments in Economics. Journal of Business & Economic Statistics 13(2): 151-161.
Nasehi SM, Sillito J, Maurer F, Burns C. 2012. What Makes a Good Code Example? A Study of Programming Q&A in StackOverflow. 2012 28th Ieee International Conference on Software Maintenance (Icsm): 25-34.
Oktay H, Taylor BJ, Jensen DD. 2010. Causal discovery in social media using quasi-experimental designs. In Proceedings of the First Workshop on Social Media Analytics. ACM.
Pan K, Kim SH, Whitehead EJ. 2009. Toward an understanding of bug fix patterns. Empirical Software Engineering 14(3): 286-315.
Parker GG, Van Alstyne MW. 2005. Two-sided network effects: A theory of information product design. Management Science 51(10): 1494-1504.
Porter MF. 1980. An Algorithm for Suffix Stripping. Program-Automated Library and Information Systems 14(3): 130-137.
Reddit. 2016. 1.7 billion reddit comments loaded on BigQuery.
Roberts JA, Hann IH, Slaughter SA. 2006. Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. Management Science 52(7): 984-999.
Shah SK. 2006. Motivation, governance, and the viability of hybrid forms in open source software development. Management Science 52(7): 1000-1014.
Stavrianou A, Andritsos P, Nicoloyannis N. 2007. Overview and semantic issues of text mining. Sigmod Record 36(3): 23-34.
Sun M, Zhu F. 2013. Ad Revenue and Content Commercialization: Evidence from Blogs. Management Science 59(10): 2314-2331.
Tiwana A. 2013. Platform Ecosystems: Aligning Architecture. Governance, and Strategy. Morgan Kaufmann Publishers Inc.: San Francisco.
Vasilescu B, Filkov V, Serebrenik A. 2013. StackOverflow and GitHub: Associations Between

Software Development and Crowdsourced Knowledge. 2013 Ase/Ieee International Conference on Social Computing (Socialcom): 188-195.
von Krogh G, von Hippel E. 2006. The promise of research on open source software. Management Science 52(7): 975-983.
Wareham J, Fox PB, Giner JLC. 2014. Technology Ecosystem Governance. Organization Science 25(4): 1195-1215.
Wasko MM, Faraj S. 2005. Why should I share? Examining social capital and knowledge contribution in electronic networks of practice. Mis Quarterly 29(1): 35-57.
Wooldridge JM. 2010. Econometric analysis of cross section and panel data. MIT Press: Cambridge.
Yang T-I, Torget AJ, Mihalcea R. 2011. Topic modeling on historical newspapers. In Proceedings of the Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities.
Yang Y, Pedersen JO. 1997. A comparative study on feature selection in text categorization. In Proceedings of the Icml.
Yoffie DB, Kwak M. 2006. With friends like these: The art of managing complementors. Harvard Business Review 84(9): 88.
Younge KA, Tong TW, Fleming L. 2015. How Anticipated Employee Mobility Affects Acquisition Likelihood: Evidence from a Natural Experiment. Strategic Management Journal 36(5): 686-708.
Zdziarski J. 2012. Hacking and securing iOS applications: stealing data, hijacking software, and how to prevent it. O'Reilly Media, Inc.
Zhu F, Iansiti M. 2012. Entry into Platform-Based Markets. Strategic Management Journal 33(1): 88-106.

Table 1. Time until jailbreak for each ios version

iOS version	iOS official release date	Jailbreak release date	Time (# days) until jailbreak
iOS 1.0	Jun-29-2007	Jul-10-2007	11
iOS 2.0	Jul-11-2008	Jul-20-2008	9
iOS 3.0	Jun-17-2009	Jun-19-2009	2
iOS 4.0	Jun-21-2010	Jun-23-2010	2
iOS 5.0	Oct-12-2011	Oct-13-2011	1
iOS 6.0	Sep-19-2012	Sep-19-2012	0
iOS 7.0	Sep-18-2013	Dec-22-2013	95
iOS 8.0	Sep-17-2014	Oct-22-2014	35

Note: This table reports the time (in number of days) between the official release date of an iOS version and the release date of jailbreak. The focal event is marked with grey shade.

Table 2. Variable definition and summary statistics

Dependent variables	Definition	Mean	S.D.	Min.	Max.
Post Count	Number of iOS or Android related online posts from a focal app developer in a given week	1.60	3.02	0	62
Post Quality	Average score of iOS or Android related online posts from a focal app developer in a given week	0.29	0.92	-2	56.67
Question Count*	Number of iOS or Android related questions posted by a focal app developer in a given week	1.13	1.69	0	31
Question Quality*	Average score of iOS or Android related questions posted by a focal app developer in a given week	0.22	1.05	-7	45
Answer Count*	Number of iOS or Android related answers posted by a focal app developer in a given week	0.15	0.42	0	10
Answer Quality*	Average score of iOS or Android related answers posted by a focal app developer in a given week	0.43	1.28	-2	72
Accepted Answer Quality*	Average score of accepted iOS or Android related answers by a focal app developer in a given week	0.95	2.76	0	177
Independent variables	**Definition**	**Mean**	**S.D.**	**Min**	**Max**
iOS	A dummy variable equal to 1 for an iOS app developer, and 0 for an Android app developer	0.33	0.47	0	1
After	A dummy variable equal to 1 for weeks after the jailbreak of iOS 7, and 0 otherwise	0.50	0.50	0	1
Question View Count*	Average of cumulative counts of views for all questions posted by a focal app developer in a given week	96.09	398.32	0	3579
Question Comment Count*	Average of cumulative counts of comments for all questions posted by a focal app developer in a given week	0.85	1.78	0	25
Edit Activity*	Number of edits proposed by a focal app developer in a given week	0.12	1.95	0	135
Comment Activity*	Number of comments proposed by a focal app developer in a given week	2.52	5.88	0	212
Add-Favorite Activity*	Number of questions added as favorite by a focal app developer in a given week	0.31	1.39	0	46
Answer Activity*	Number of answers proposed by a focal app developer in a given week	1.55	2.67	0	49

Note: Please refer to the Data and Methods section for detailed operationalization. The values of the descriptive statistics reported in the table are based on the StackOverflow.com sample. Variables indicated by an asterisk are only available for the StackOverflow.com sample, while the other variables are available for both the StackOverflow.com sample and Reddit.com sample.

Table 3. Comparison of the treatment sample and control sample (StackOverflow.com sample)

Panel A: Technology tags of app developers

Names of shared tags	Names of iOS-unique tags	Names of android-unique tags
javascript	ios	android
java	uiviewcontroller	android-intent
c#	ios5	android-activity
jquery	automatic-ref-counting	android-listview
php	ios6	actionbarsherlock
html	nsstring	opengl
c++	ios4	nullpointerexception
css	autolayout	google-cloud-messaging
python	storyboard	spring-mvc
c	cocos2d-iphone	guava
ruby-on-rails	core-animation	user-interface
.net	uibutton	android-edittext
ruby	uicollectionview	zend-framework
arrays	sprite-kit	gradle
mysql	uikit	libgdx

Note: Panel A lists the most frequently-assigned technology tags (ordered by frequency) shared by the iOS and Android developers, or unique to the iOS or Android developers.

Panel B: Characteristics of knowledge sharing before and after jailbreak

	Before jailbreak			After jailbreak			
Variables	iOS mean	Android mean	Diff before	iOS mean	Android mean	Diff after	Univariate DD (t-stat)
Post Count	1.80	1.70	**0.10**	1.45	1.48	**-0.03**	**-0.13 (-1.93)**
Post Quality	0.42	0.31	**0.11**	0.24	0.24	**0.00**	**-0.11 (-4.95)**
Question Count	1.21	1.19	**0.02**	1.06	1.06	**0.00**	**-0.02 (-0.85)**
Question Quality	0.34	0.22	**0.12**	0.17	0.17	**0.00**	**-0.12 (-5.10)**
Answer Count	0.19	0.16	**0.03**	0.14	0.13	**0.01**	**-0.02 (-2.89)**
Answer Quality	0.60	0.45	**0.15**	0.38	0.36	**0.02**	**-0.13 (-4.38)**
Accepted Answer Quality	1.32	1.01	**0.31**	0.80	0.80	**0.00**	**-0.31 (-4.96)**

Note: Panel B provides a summary of mean comparisons of knowledge sharing between the iOS and Android developers *before* and *after* the jailbreak of iOS 7. Differences between the iOS and Android developers *before* and *after* the jailbreak, as well as the univariate diff-in-diffs (last column) are bolded.

Panel C: Covariate balance before jailbreak

Variables	iOS mean	Android mean	Difference (t-stat)
Question View Count	107.77	90.28	17.49 (1.73)
Question Comment Count	0.82	0.86	-0.04 (-1.58)
Edit Activity	0.10	0.13	-0.03 (-1.29)
Comment Activity	2.63	2.46	0.17 (1.53)
Add-Favorite Activity	0.29	0.32	-0.03 (-1.32)
Answer Activity	1.54	1.56	-0.02 (-0.24)

Note: Panel C provides a summary of mean comparisons of covariates between the iOS and Android developers *before* the jailbreak.

Table 4. Difference-in-differences results for aggregated knowledge sharing (StackOverflow.com sample and Reddit.com sample)

Variables	StackOverflow.com sample		Reddit.com sample	
	(1)	(2)	(3)	(4)
	Post Count	Post Quality	Post Count	Post Quality
DD (iOS*After)	-0.16	-0.07	-0.04	-0.04
	(0.04)	(0.02)	(0.02)	(0.02)
Question View Count	0.00	0.00		
	(0.00)	(0.00)		
Question Comment Count	-0.02	0.05		
	(0.01)	(0.01)		
Edit Activity	-0.02	0.00		
	(0.00)	(0.00)		
Comment Activity	0.05	-0.00		
	(0.01)	(0.00)		
Add-Favorite Activity	0.05	0.01		
	(0.02)	(0.00)		
Answer Activity	0.84	0.02		
	(0.02)	(0.00)		
Constant	0.12	0.11	0.36	0.27
	(0.04)	(0.04)	(0.03)	(0.02)
Individual FE	Yes	Yes	Yes	Yes
Week FE	Yes	Yes	Yes	Yes
R-squared	0.65	0.21	0.21	0.16
No. of Observations	33,993	33,993	13,905	13,905
No. of Users	1,259	1,259	515	515

Note: This table reports difference-in-differences results using the StackOverflow.com sample (Columns 1-2) and Reddit.com sample (Columns 3-4). Dependent variables are *Post Count and Post Quality* as defined in Table 2. All RHS variables are defined as in Table 2. The unit of analysis is the individual-week level. All models include individual fixed effects and week fixed effects. In all columns, the dummy variables *iOS* and *After* are dropped out because of perfect collinearity with the fixed effects. Robust standard errors are reported in parentheses. p-values for DD terms are reported in the text. Two-tailed tests.

Table 5. Difference-in-differences results for various steps of knowledge sharing (StackOverflow.com sample)

Variables	(1) Question Count	(2) Question Quality	(3) Answer Count	(4) Answer Quality	(5) Accepted Answer Quality
DD (iOS*After)	-0.04	-0.09	-0.02	-0.10	-0.23
	(0.02)	(0.02)	(0.01)	(0.03)	(0.05)
Question View Count	0.00	0.00	0.00	0.00	0.00
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Question Comment Count	0.04	0.06	0.01	0.05	0.06
	(0.00)	(0.01)	(0.00)	(0.01)	(0.02)
Edit Activity	-0.01	0.00	0.00	0.00	0.00
	(0.00)	(0.00)	(0.00)	(0.00)	(0.01)
Comment Activity	0.03	0.00	0.00	0.00	-0.00
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Add-Favorite Activity	0.04	0.02	0.01	0.02	0.04
	(0.01)	(0.01)	(0.00)	(0.00)	(0.01)
Answer Activity	0.48	0.00	0.06	0.05	0.33
	(0.01)	(0.01)	(0.00)	(0.01)	(0.02)
Constant	0.20	0.01	0.01	0.19	0.12
	(0.02)	(0.04)	(0.01)	(0.05)	(0.10)
Individual FE	Yes	Yes	Yes	Yes	Yes
Week FE	Yes	Yes	Yes	Yes	Yes
R-squared	0.71	0.19	0.27	0.13	0.29
No. of Observations	33,993	33,993	33,993	33,993	33,993
No. of Users	1,259	1,259	1,259	1,259	1,259

Note: This table reports difference-in-differences results for various steps involved in knowledge sharing using the StackOverflow.com sample. Dependent variables are *Question Count*, *Question Quality*, *Answer Count*, *Answer Quality*, and *Accepted Answer Quality*, as defined in Table 2. All RHS variables are defined as in Table 2. The unit of analysis is the individual-week level. All models include individual fixed effects and week fixed effects. In all columns, the dummy variables *iOS* and *After* are dropped out because of perfect collinearity with the fixed effects. Robust standard errors are reported in parentheses. p-values for DD terms are reported in the text. Two-tailed tests.

Table 6. Robustness test: CEM matched sample (StackOverflow.com sample)

Variables	(1) Post Count	(2) Post Quality	(3) Question Count	(4) Question Quality	(5) Answer Count	(6) Answer Quality	(7) Accepted Answer Quality
DD (iOS*After)	-0.17	-0.08	-0.05	-0.10	-0.02	-0.10	-0.25
	(0.04)	(0.02)	(0.02)	(0.02)	(0.01)	(0.03)	(0.05)
Question View Count	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Question Comment Count	-0.02	0.05	0.04	0.06	0.01	0.05	0.05
	(0.01)	(0.01)	(0.00)	(0.01)	(0.00)	(0.01)	(0.02)
Edit Activity	-0.02	0.01	-0.01	0.01	0.00	0.00	0.01
	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.01)
Comment Activity	0.06	-0.00	0.04	-0.00	0.00	-0.00	-0.01
	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
Add-Favorite Activity	0.07	0.01	0.05	0.02	0.01	0.02	0.04
	(0.02)	(0.00)	(0.01)	(0.01)	(0.00)	(0.01)	(0.01)
Answer Activity	0.82	0.03	0.48	0.01	0.06	0.05	0.34
	(0.02)	(0.00)	(0.01)	(0.01)	(0.00)	(0.01)	(0.02)
Constant	0.12	0.11	0.20	0.01	0.01	0.19	0.12
	(0.04)	(0.05)	(0.02)	(0.04)	(0.01)	(0.06)	(0.11)
Individual FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Week FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
R-squared	0.65	0.22	0.71	0.20	0.26	0.13	0.31
No. of Observations	32,184	32,184	32,184	32,184	32,184	32,184	32,184
No. of Users	1,192	1,192	1,192	1,192	1,192	1,192	1,192

Note: This table reports results of robustness checks using the StackOverflow.com sample with coarsened exact matching (CEM). Model specifications are the same as the main analyses in Table 4 and Table 5. All models include individual fixed effects and week fixed effects. In all columns, the dummy variables *iOS* and *After* are dropped out because of perfect collinearity with the fixed effects. Robust standard errors are reported in parentheses. Two-tailed tests.

Table 7. Robustness test: users of alternative activity levels (stackoverflow.com sample)

Activity Level	Variables	Post Count	Post Quality	Question Count	Question Quality	Answer Count	Answer Quality	Accepted Answer Quality
5%	DD (iOS*After)	-0.17 (0.05)	-0.08 (0.02)	-0.05 (0.02)	-0.08 (0.02)	-0.02 (0.01)	-0.09 (0.03)	-0.24 (0.06)
10%	DD (iOS*After)	-0.16 (0.04)	-0.07 (0.02)	-0.04 (0.02)	-0.08 (0.02)	-0.02 (0.01)	-0.09 (0.03)	-0.23 (0.05)
15%	DD (iOS*After)	-0.13 (0.04)	-0.07 (0.02)	-0.04 (0.01)	-0.08 (0.02)	-0.02 (0.01)	-0.10 (0.03)	-0.22 (0.05)
20%	DD (iOS*After)	-0.16 (0.04)	-0.07 (0.02)	-0.04 (0.02)	-0.09 (0.02)	-0.02 (0.01)	-0.10 (0.03)	-0.23 (0.05)
	Controls and Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Note: This table summarizes results of robustness checks using three samples of app developers with different activity levels (5%, 10%, and 15%, respectively) on StackOverflow.com. Model specifications are the same as the main analyses. Robust standard errors are reported in parentheses. Two-tailed tests.

Table 8. Robustness test: windows phone app developers as an alternative control group (stackoverflow.com sample)

Variables	(1) Post Count	(2) Post Quality	(3) Question Count	(4) Question Quality	(5) Answer Count	(6) Answer Quality	(7) Accepted Answer Quality
DD (iOS*After)	-0.22 (0.07)	-0.11 (0.06)	-0.07 (0.04)	-0.11 (0.06)	-0.07 (0.01)	-0.17 (0.10)	-0.37 (0.19)
Question View Count	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
Question Comment Count	-0.01 (0.01)	0.06 (0.01)	0.03 (0.01)	0.06 (0.01)	0.01 (0.00)	0.04 (0.01)	0.00 (0.03)
Edit Activity	-0.03 (0.01)	-0.01 (0.00)	-0.02 (0.01)	-0.00 (0.01)	-0.00 (0.00)	-0.01 (0.00)	-0.02 (0.03)
Comment Activity	0.07 (0.01)	0.00 (0.00)	0.04 (0.01)	-0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.01 (0.01)
Add-Favorite Activity	0.09 (0.02)	0.01 (0.01)	0.05 (0.01)	0.00 (0.01)	0.01 (0.00)	0.01 (0.01)	0.03 (0.02)
Answer Activity	0.77 (0.02)	0.03 (0.00)	0.47 (0.01)	0.00 (0.01)	0.05 (0.00)	0.05 (0.01)	0.36 (0.02)
Constant	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Individual FE	0.25	0.20	0.23	0.04	0.05	0.36	0.25
Week FE	(0.06)	(0.05)	(0.03)	(0.06)	(0.02)	(0.11)	(0.19)
R-squared	0.62	0.32	0.71	0.37	0.16	0.12	0.37
No. of Observations	17,253	17,253	17,253	17,253	17,253	17,253	17,253
No. of Users	637	637	637	637	637	637	637

Note: This table reports results of robustness checks using Windows Phone OS app developers as an alternative control group, based on the StackOverflow.com sample. Model specifications are the same as the main analyses. All models include individual fixed effects and week fixed effects. In all columns, the dummy variables *iOS* and *After* are dropped out because of perfect collinearity with the fixed effects. Robust standard errors are reported in parentheses. Two-tailed tests.

Figure 1. Graphical illustration of the research design

Note: Figure 1 depicts the quasi-experimental research design. Before jailbreak (the treatment), iOS app developers ("treatment group" on the top left part) experience strong gatekeeping by Apple. By contrast, Android app developers ("control group" on the bottom left part) do not experience such gatekeeping by Apple. After the treatment (jailbreak of iOS 7), jailbreak app developers may join iOS app developers' community (on the top right part), potentially changing the size and composition of complementors of iOS. By contrast, Android app developers (on the bottom right part) should not experience such changes because the jailbreak does not affect them.

Figure 2. Topic clouds before and after jailbreak

Note: Figure 2 shows the topic clouds (using information gain and Latent Dirichlet Allocation) based on all jailbreak related posts by on Reddit.com. Before the jailbreak of iOS 7, jailbreak developers tend to discuss technical issues related to general jailbreak process or jailbreak specific tweak development, including topics such as tweak or activator. After the jailbreak, there is a noticeable increase in the discussion of iOS related app development, including topics such as ios7 or iphone. This comparison of word clouds indicates that some jailbreak app developers shift their attention from jailbreak development to iOS related app development after the jailbreak, leading to a change in the composition of complementors of iOS.