The Ecosystem of Software Platform: A Study of Asymmetric Cross-side Network Effects and Platform Governance

Peijian Song
Nanjing University
songpeijian@nju.edu.cn

Ling Xue
Georgia State University
lingxue.xue@gmail.com

Arun Rai
Georgia State Univerity
arunrai@gsu.edu

Cheng Zhang
Fudan University
zhangche@fudan.edu.cn

Abstract: While past research on software platform has recognized the existence of cross-side network effects (CNEs) between the application side and the user side, little is known about the asymmetry between the CNEs of the two sides on each other. Informed by a perspective of complex adaptive systems, this study theorizes how the user-to-application CNE is temporally different from the application-to-user CNE, and how these CNEs may be influenced by the governance mechanisms of the platform. We empirically test our theoretical arguments using a longitudinal data about a leading software browser. Our first main finding is the temporal asymmetry between the user-to-application CNE and the application-to-user CNE. Specifically, while the increased installed base of end users has a primarily long-term impact on the growth of application number and variety, the increased number and variety of applications have primarily short-term impacts on the growth of end users. Our second finding is that the length of application review time weakens the long-term user-to-application effect, but not the short-term application-to-user effect. Third, we also find that frequent platform updating can significantly weaken both the long-term user-to-application CNE and the short-term application-to-user CNE. Our study generates important theoretical and practical implications.

Keywords: Software platform, two-sided market, network effect, IS governance, complex adaptive systems

The Ecosystem of Software Platform: A Study of Asymmetric Cross-side Network Effects and Platform Governance

1. INTRODUCTION

Software platforms, such as Apple's iOS and Google's Android, are emerging as dominant models for software-based services (Evans et al. 2006; Tiwana et al. 2010). Unlike standalone application systems, software platforms are extensible codebases of software systems that provide core functionalities for applications that run on them (Baldwin and Woodard 2009). Based on their technological architectures, platforms extend their product boundaries by attracting large numbers of third-party applications that create complementary value (Boudreau 2012; Ceccagnoli et al. 2012). In this way, software platforms inherently operate as two-sided markets and exhibit cross-side network effects (CNE) (Anderson et al. 2014; Zhang et al. 2012), i.e., the users on one side and the applications (offered by third-party developers) on the other side influence the growth of installed bases of each other. The installed base in the context of platforms is typically assessed in terms of number of applications, application variety, and users' usage of applications, etc.

CNEs have strategic implications for platform ecosystems. First, CNEs significantly influence a platform's strategies to build the installed base on either side. For instance, a platform may sacrifice profits on one side to build the installed base and make the ecosystem more appealing on the other side (Hagiu 2006; Parker and Van Alstyne 2005; Weyl 2010). Second, CNEs can help market incumbents to achieve critical advantage by allowing them to leverage their market sizes on one side to build dominant positions on the other side, creating substantial entry barriers for new comers. A software platform that starts with minor leads on both sides is likely to win the entire market over time, even if it may be inferior in quality (Kim et al. 2013; Lieberman 2007; Zhu and Iansiti 2012).

Therefore, understanding the mutual influences between the two sides of a platform through CNEs is critical for the software platform to establish competitive advantage using such effects.

Past research has focused primarily on the existance of CNEs in the two-sided markets. Although the potential difference in the influences between the two sides have received some recent attention in other contexts, e.g., television networks (Wilbur 2008), little is known about the asymmetric interactions between the two sides of the software platform through CNEs.[1] The major participants on the two sides of software platform, i.e., individual users and application developers, are featured by very different objectives, behavior, and capabilities. Although the two sides may reinforce each other through CNEs, there can be differences in their mutual influences arising from the distinct features of each of the two sides. These resulting asymmetric influences of CNEs between the two sides, if present, should have important implications for the platform's success on both sides of the market. This issue of asymmetric CNEs in software platforms, however, has not been adequately investigated in the existing literature.

Another important and related issue is how platform governance influences CNEs in software platforms. Platform governance refers to the policies and mechanisms that the platform adopts to govern its operations on the two sides and maintain its ecosystem (Tiwana et al. 2010). For example, a software platform can issue development guidelines and provide standard development and testing kits for third-party applications (Ghazawneh and Henfridsson 2013). The platform can also review all applications to ensure that they meet certain criteria before making them available to the public

[1] For example, Wilbur (2008) examines television networks and shows that while the viewer base helps attract more advertisers, advertising negatively affects the viewer side. However, in this setting, the viewer side is featured by the demand for programs rather than advertisements. In contrast, in the setting of software platform with CNEs, the user side is also featured by the demand for third-party applications offered by the application side.

(Maurer and Tiwana 2012). The governance policies of the platform influence the behaviors of end users, third-party application developers, and their mutual influences. Therefore, it is worthwhile to examine how platform governance influences CNEs, and potentially affects the user side and the application side differently. Although the relationship between platform governance and the evolving dynamics of CNEs has received some consideration (e,g, Tiwana et al. 2010), the relationship requires conceptual elaboration and empirical investigation. Our research also addresses this gap.

Our research focuses on two research questions, which are motivated by the complex adaptive business systems (CABS) theoretical perspective (McKelvey 1997; Tanriverdi et al. 2010). The first research question is how the CNE of the application side on the user side (referred to as *application-to-user effect* from now on) differ from the CNE of the user side on the application side (referred to as *user-to-application effect*)? Informed by a CABS perspective, we suggest that although the two sides of a software platform are interconnected, they are likely to exhibit different interdependencies, specifically with respect to the timing and duration of the *application-to-user* and *user-to-application* effects. These complex dynamics in interdependence cannot be sufficiently explained by traditional IS theories that tends to be static and mechanistic. Using longitudinal data about a major software platform, we employ a vector-autoregressive analysis (VARX) to examine the dynamics of CNEs in terms of both the *short-term effect* and the *long-term effect*. The short-term effect reflects the immediate but potentially transient impact that one side may have on the growth of installed base on the other side. The long-term effect reflects the potentially lagged but more stabilized impact that one side may have on the other (Chang and Gurbaxani 2012; Pauwels and Weiss 2008). The consideration of the distinction between the two CNEs and their respective short-term and long-term effects enables us to elaborate our understanding about the interdependence between the two sides of a market in a platform ecosystem.

The second research question we consider is how platform governance influences the application-to-user CNE and the user-to-application CNE, potentially in different ways. From a CABS perspective (Allen and Varga. 2006; Benbya and McKelvey 2006; Tanriverdi et al. 2010), effective platform governance is about establishing mechanisms that enable adaptiveness of the ecosystem and requires understanding the nature of interdependencies between the two sides of the market. We specifically focus on two key aspects of software platform governance, i.e., application review time and platform updating frequency, and examine their impacts on the interdependencies between the two sides of the market. Application review time refers to the time it takes for a platform to examine third-party applications and verify that they comply with platform requirements (before making them available to users). Application review time influences CNEs because it not only influences the development plans of third-party applications developers, but also influences how often individual users can access new applications. Platform updating frequency refers to how often the platform is improved through new versions and functionalities (Ji et al. 2005; Khoo and Robey 2007). Platform updating frequency influences both the user side and the application side, as both sides need to adapt to platform updates. The consideration of how these platform governance strategies affect the interdependence between the two sides of the market helps us better understand how a platform and the two sides of the market can adapt and co-evolve.

We conduct an empirical study using a unique dataset about a leading web browser: Mozilla Firefox. Introduced in November 2004, Firefox is built upon a model of open-source platform ecosystem that provides development kits and application programming interfaces to support third-party applications. We construct a longitudinal dataset covering a period from 2008 to 2013.

This dataset includes weekly data on activities of both end users and third-party application developers, and Firefox's application review time and updating frequency. Using vector autoregressive analysis (VARX), we examine the dynamics of CNEs between the application side and the user side, and how they are influenced by Firefox's application review time and updating frequency.

Our analysis generates several important findings. First, we show the temporal asymmetry between the application-to-user effect and the user-to-application effect. Specifically, both the increased quantity and the increased variety of applications drive the platform usage of end users in the short term. In contrast, the increased platform usage of end users drives the growth of application quantity and variety in the long term. Second, we find that the length of applications review time influences the application side and the user side differently. Although it significantly weakens the long-term user-to-application effect, its impact on the short-term application-to-user effect is not significant in general. Third, we find that the frequency of platform updating significantly weakens the long-term user-to-application effect. However, regarding the short-term application-to-user effect, the frequency of platform updating significantly weakens the effect of application quantity, but not that of application variety, on the platform usage of end users.

This study generates several key contributions. First, departing from the existing literature that focuses on the presence of CNEs (e.g. Park 2004) and the impact of CNEs on platform competitive advantage (e.g., Kim et al. 2013; Zhu and Iansiti 2012), this study illuminates the difference between the application-to-user and the user-to-application network effects. The results help develop better understanding of the temporal dynamics underlying the interdependence between the two sides of the platform's markets. Second, while the past research has primarily developed theoretical understanding of the platform ecosystem (e.g., Ceccagnoli et al. 2012; Parker and Van Alstyne 2008; Tiwana et al. 2010), our study empirically illustrates how the governance mechanisms of a platform itself can influence the interactions between different parties in this ecosystem. These findings generate important implications regarding how the platform may improve its management of the ecosystem. Third, our study also contributes to the general IS strategy and governance literature (e.g., Tanriverdi et al. 2010; Tiwana et al. 2010) by illustrating the importance for platforms to develop governance strategies to adapt to the complex, dynamic environment with temporary and lasting features of evolution.

2. THEORETICAL AND HYPOTHESIS DEVELOPMENT

2.1 Software Platform and Two-Sided Markets

A software platform manifests itself as an ecosystem that encompasses several distinct players, including (1) end users of the platform and associated applications; (2) third-party developers who provide applications that are used along with the platform; and (3) the platform that bridges end users and third-party applications (Cusumano and Gawer 2002; Eisenmann et al. 2010; 2011). A platform's performance is contingent not only on its interactions with both the user side and the application side in two-sided markets but also on the interdependencies between these two sides through CNEs (Boudreau 2012; Parker and Van Alstyne 2008). Given the diverse players involved in a platform ecosystem, a complex adaptive business system perspective suggests that achieving adaptation and co-evolution across the user and application sides of the market requires developing a nuanced understanding of their interdependence (Tanriverdi et al. 2010). And, developing this understanding requires uncovering the temporal dynamics underlying how each side of the market adapts to changes on the other side and how the two sides co-evolve.

To examine CNEs, prior research on two-sided markets has primarily shown the existence and magnitude of CNEs in the platform business model (e.g. Park 2004). However, less attention has been paid to the potential difference in the nature of the independence between the two sides, especially the temporally asymmetry between the user-to-application effect and the application-to-user effect. Stremersch et al. (2007) investigate the temporally asymmetric pattern of indirect network effects in consumer electronics market, and find that while hardware sales drive software availability, there lacks the opposite effect. The temporal asymmetry in CNEs, however, is distinct in two important ways from that in indirect network effects. First, indirect network effects primarily concern the influences between different products that are provided to the same user base (Basu et al. 2003; Stremersch et al. 2007). In contrast, CNEs concern the mutual influences between users and applications though the platform. The asymmetric mutual influences of users and applications on each other should have more implications for the platform which interacts with both of them. Second, studies on indirect network effects do not have to consider how these effects may be influenced by a mediator. In CNEs, however, the platform acts as the key market maker to bridge end users and applications (Spulber 2010). From a CABS perspective, the consideration of the mediator role played by the platform is also critical for understanding how to manage the asymmetric interdependence between the two sides, their adaptation to changes on the other side, and their co-evolution.

In prior research, attention has been paid to how the platform may benefit from CNEs (e.g., in gaining competitive advantage) (Chen and Xie 2007; Kim et al. 2013; Zhu and Iansiti 2012), rather than to how the platform may influence CNEs through its governance mechanisms. Software platforms adopt various policies and activities in governing and operating their ecosystems (Tiwana et al. 2010). Past research has generally recognized the potential impact of platform governance on the evolving dynamics of ecosystems (e.g. Parker and Van Alstyne 2008; Tiwana et al. 2010). However, there is the need to theorize and empirically test the influence of specific platform governance mechanism on CNEs.

2.2 Value Creation and Value Capture in Cross-Side Network Effects

Network effects are characterized by two key aspects: *value creation* and *value capture* (Afuah 2013). Value creation refers to the processes that networks create value for their participants (Lepak et al. 2007), and this value is influenced by various network aspects such as network size, network structure, and network conduct. Value capture refers to the value appropriation processes in which network members derive or retain the value (Bowman and Ambrosini 2000). Value creation and value capture should be viewed as distinct processes in the network environment, since the value created from one source may not necessarily be captured by other network participants (Bowman and Ambrosini 2000; Chatain and Zemsky 2011). This distinction between value creation and value capture can be useful to understand how governance mechanisms can enable a firm to manage its co-evolution along with its partners (suppliers, customers) (Rai and Tang 2014).

Regarding CNEs in the software platform ecosystem, both the user side and the application side can create value for the other side (Spulber 2010). End users can potentially derive utilities from using third-party applications, and application developers may gain financially or in reputation when end users adopt their products. However, both sides need to take effort to identify and capture the value created by the other side, the processes of which may not be trivial (Lepak et al. 2007). Application developers need to study users' preferences and identify their application needs. Resource commitment is required in application development before developers profit from user demand (Ghazawneh and Henfridsson 2013). Once their applications are available to end users, developers also face challenges of competition and imitation, which cause more uncertainty for value capture (Boudreau 2012). On the user side, end users need to sample competing applications and make usage decisions. Sometimes they also have to adapt to the new or updated versions of applications. The differences in the interdependencies between application developers and end users can arise from their different processes of value creation and value capture. Such differences in interdependencies are likely to be featured by the two sides' differential responses to each other in terms of immediacy and duration. One side may respond more quickly to the other side, and the response of one side to the other side may be more durable than the reverse response. This difference can be captured by the difference between the short-term and the long-term CNEs on either side of the platform.

2.3 Temporal Asymmetry in Cross-Side Network Effects

Due to the nature of value creation and value capture on the application side, the response of the application side to the increased installed base on the user side may not be instantaneous. But this response is likely to be lasting, resulting in a long-term user-to-application effect. First, the installed base of platform users is not automatically converted to the demand for certain third-party applications. From the value creation perspective, developers still need to identify the users' various needs for complementary functionalities and it often takes time for these needs to be revealed. In addition, developers are usually heterogeneous in their capabilities to discover user needs (Boudreau 2010; Haigu 2006). As a consequence, the increased user base of the platform is likely to result in a smooth and steady increase of applications over time, rather than transient hikes in application supply.

Second, from the value capture perspective, developers need to commit effort and resources to provide the applications and meet user requirements, the processes of which are often time-consuming (Boudreau 2012). For instance, even with development kits and application programming interfaces

(APIs) provided by software platforms, a fundamental application may still take at least a couple of weeks in development (Ghazawneh and Henfridsson 2013). In this regard, the application side is less likely to grow as an instantaneous response to the user side. Developers are also heterogeneous in their development capabilities, which results in variant release time and continuous growth in application supply. Third, third-party applications often need to pass the platform's review before they are available to end users. The review processes, which are not under developers' controls (Maurer and Tiwana 2012), can further smoothen the release of new applications and result in steady increases of applications over time rather than transient hikes in application supply. Therefore, the user-to-application effect is likely to exhibit as a long-term (i.e., less instantaneous but more durable) effect.

In contrast, the application-to-user effect is likely to exhibit as a short-term effect as the response of the user side to the application side is often immediate and transient. End users usually need less resource commitments to try applications, Software platforms also provide many user-friendly features to further facilitate the value capturing processes of end users. For example, applications centers and search functions offered by software platforms usually allow end users to easily locate, download and try applications that fit their needs (Haigu and Spulber 2013). From the value creation perspective, end users can also easily generate online word-on-mouth that quickly increases the demand of certain applications (Zhang et al. 2012). In this regard, the user side is likely to respond quickly to the application side with a growth. However, such instantaneous growth may not be sustainable in the long-run. Users may use free applications as a way to discover their needs, which does not necessarily lead to stable usage over time. User-friendly features of the platform can also result in low switching costs for users, making users less committed to their usage (Haigu 2006). In this regard, the growth in the application side is likely to trigger a series of temporary usage spikes that are less stabilized over time. We therefore develop the following hypothesis:

H1: *The user-to-application effect is primarily a long-term effect and the application-to-user effect is primarily a short-term effect.*

2.4 Application Review

Applications review is one of the most important platform control mechanisms (Ghazawneh and Henfridsson 2013; Maurer and Tiwana 2012). Platforms often impose standards on the content and technical specifications of third-party applications. The review processes ensure that third-party applications perform as expected, and are reliable and free of inappropriate features. For example, Apple's App Store imposes a wide range of rules covering a variety of aspects from user interface design, functionality, content, to the use of specific technologies. The application review by App Store can take up to 2 weeks. Submitted apps can be rejected for a variety of reasons, such as lack of information, bugginess and poor interface (Starr 2014). Developers, when failing to pass the application review, have to fix the problems before resubmitting their applications to the App Store. Reviews on the revised versions further delay the final application releases.

The application review time has a direct influence on application development. Developers have to take into account possible delays in their application releases. Time-consuming review processes prevent developers from quickly responding to the increased demand from the user side. In addition, the slow review processes also hurt application developers' investment incentives (Hilkert et al. 2010). Developers may forgo the opportunities of tapping into the user base of a platform with a less efficient review process and switch to other competing platforms. Therefore, we expect that long application review time weakens the user-to-application effect.

Likewise, time-consuming review processes are likely to hinder the response of the user side to the application side. As users' tastes are volatile in the online world (Sun 2012), long application review time may eventually make approved applications more obsolete and less appealing to end users compared to similar applications on other competing platforms with shorter application review time. From the value creation perspective, the increased supply of less appealing applications does not necessarily lead to substantial growth on the user side of the platform. We therefore expect that the application-to-user effect is also weakened by long application review time. We have the following hypotheses:

H2a: *Long application review time weakens the user-to-application effect.*

H2b*: Long application review time weakens the application-to-user effect.*

2.5 Platform Updating

Platform updating is an important way that a software platform enhances its own service (Tiwana et al. 2010), and changes how the interdependencies among diverse parties in the platform ecosystem are managed (Tanriverdi et al. 2010). A key feature determining the evolving dynamics of platform ecosystem is platform updating frequency, or, how often an updated version of the platform is released. Since the platform acts as the basis for third-party applications and end users to interact with each other, platform updating frequency should influence both the temporality of the user-to-application effect and the application-to-user effect. Frequent platform updates drive application developers to constantly update their products. However, application developers are heterogeneous in their capabilities to respond in a timely manner to platform updates and keep their applications up-to-date (Boudreau 2010; Haigu 2006). Consequently, frequent platform updates prevent more application developers from reacting to the increased platform user base by creating timely applications and capturing the demand from the user side. Moreover, frequent platform updating may force developers to commit more to updating their existing applications to keep their current users, increasing the difficulty for them to develop new applications to capture more value from the increased demand. Therefore, frequent platform updating is likely to weaken the user-to-application effect.

From the perspective of end users, frequent platform updating may constrain their intentions and capabilities to capture the value of increased applications. It is often burdensome for individual users to frequently upgrade their installed software platforms. When frequent platform updating only brings marginal benefits, some users may defer their upgrading movements. Consequently, frequent platform updating discourages certain end users from using more up-to-date applications, resulting in less cross-side spillover effect of the increased base of applications. End users may also have insufficient cognitive resources to keep up with a rapid pace of platform updates. Once they miss some of the platform upgrades that are introduced in a short time window, end users may lag in adapting to the new platform and gradually lose the momentum of using the software platform (Venkatesh et al. 2008). The usage barrier caused by platform updates can also weaken the response of the user side by reducing externalities within users and undermines the user-side value creation. Therefore, frequent platform updating is likely to weaken the application-to-user effect. We develop the following hypotheses:

H3a: *Frequent platform updating weakens the user-to-application effect.*

H3b*: Frequent platform updating weakens the application-to-user effect.*

3. METHODOLOGY

3.1 Research Context

Our empirical setting is a major web browser system – Mozilla’s Firefox. Since its debut in late 2004, Firefox had been the second-most used web browser until late 2011 (when Google Chrome surpassed it). Nowadays it is the third most widely used web browser, with approximately 20% in market share. As an open-source system, Firefox relies more on a platform ecosystem to sustain its operations. It extends its product boundaries by encouraging a large number of third-party developers to supply complementary applications. These applications offer add-on features, such as videoconferencing and privacy protection, based on Firefox's core functionalities.

To ensure application quality and user experience, Firefox mandates that all hosted applications should be reviewed by a team of editors. Developers are required to upload their application files to the developer hub, and provide related information such as descriptions and preview images. Uploaded application files are then scanned by editors using a variety of tools, which can warn the editors about potential flaws in the source code. Developers may be asked (usually through emails) to provide additional information or improve certain aspects of the applications before they are further evaluated. The review notes may be used in the evaluation of resubmitted applications that have been rejected before.

Firefox also updates its own browser platform to add features and improve quality. These updates may also require third-party applications to be updated accordingly for compatibility with the browser platform. For end users, Firefox can automatically check for the availability of updates and inform them to manually upgrade their installed browsers to the latest version. The upgrading, however, may require time-consuming processes of downloading and rebooting of users' computers. From late 2004 to early 2011, Firefox updated its browser platform approximately once a year. In early 2011, Firefox changed its platform update policies and the average time intervals between updates were reduced sharply to six weeks.

The platform ecosystem of Firefox offers an ideal setting for our empirical analysis. First, as Firefox is free to end users, the examination of CNEs is not likely to be confounded by any pricing effect. Second, the editors of Firefox platform offer weekly application review data through the application forum. The longitudinal data collected from the applications forum can be used to assess the frequencies and duration of application reviews and to quantify their effects on CNEs. Finally, Firefox's platform updating policies are largely exogenous to CNEs, as Firefox's updating decisions are generally not based on the interactions between the application side and the user side. This feature should minimize the concerns about endogenous updating decisions and enable us to better assess the causal effect of platform updating on CNEs.

3.2 Data and Measurement

We test our hypotheses using a longitudinal weekly dataset about the Firefox ecosystem from July 2009 to October 2013 (223 weeks). We choose weekly data for two reasons. First, daily data about CNEs and platform governance does not reveal sufficient variation, and monthly data does not provide the granularity to reveal the ongoing patterns of CNEs and platform governance. Second, our data indicates that the status of third-party applications often changes on a weekly basis.

We collected information about end users and third-party applications from the official website for Firefox add-on applications (https://addons.mazilla.org/en-us/firefox). This site records all third-party add-on applications for Firefox since it started in 2004. For each application, it provides detailed information, including the initial launch time, updating history (i.e., when and how many times), category, and the developer's name. These applications run on Firefox, and are free to end users as complementary components of the Firefox browser platform. As of October 2013, there were more than 10,000 applications available on this platform, spanning 14 categories.

We collected longitudinal data on application releases and updates and aggregated it at the week level. In examining CNEs, we focus on two key aspects of the application side: the quantity of applications, and the variety of applications. The consideration of application quantity helps reflect the direct CNEs through which the increased user base drives the growth of overall application usage. The consideration of application quantity further helps better reflect the indirect CNEs through which the increased user base drives the usage of more complementary applications.

To capture quantity, we use a measure of *the total number of applications* (NA) that are available on the Firefox platform every week. We log-transform this quantity measure to address the skewness in distributions. To capture application variety, we use a measure of *the diversity of applications* (DA). The measure is similar in nature to the Herfindahl index of concentration/ diversification. Specifically, the Herfindahl index of a platform is represented as $HI = \sum_{i=1}^{N}(x_i/x)$, where x_i is the number of applications in the *i*-th category and x is the total number of applications. x_i / x is therefore the share of the *i*-th category of applications. A lower level of weekly Herfindahl index indicates a higher degree of application diversity (variety) in that week. To make the result interpretation more intuitive, we measure DE by subtracting the weekly Herfindahl index value from 1 (i.e. maximum Herfindahl index value). Consequently, a higher DE value means a higher level of weekly application variety.

Regarding the user side, we consider a measure of *user usage* (UU) defined as the average daily times of actual usage (of the Firefox browser) by end users on a weekly basis. This measure is based on Firefox's tracking of every user's usage behavior. Therefore, UU provides more objective and accurate information about user demand than other alternative measures, such as downloads.

In examining the applications review time, we collected application review data from the official Firefox application forum (https://forums.mozilla.org/addons/), where the editors of Firefox post the weekly status of the application review queue. We use this data to calculate the average waiting time of applications, and use the average waiting time as a measure of the *applications review time* (AR). Finally, our measure of *platform updating frequency* (PU) is a binary indicator with a value of 0 for all weeks before March 21, 2011 (when Firefox changed its policy to update itself more frequently), and a value of 1 for all weeks after March 21, 2011. Our entire sample covers 89 pre-change weeks and post-change 123 weeks.

We also control for three exogenous factors. The first one is the *entry of the key competitor (CE)*, which may affect the platform's performance on both sides. CE is measured using a dummy variable indicating whether or not the key competitor, Google Chrome, had entered the browser market. The second factor is *platform market share (PMS)*, which may influence both the application side and the user side on an ongoing basis. PMS is measured using the proportion of all web browser users who use Firefox, and this data was obtained from StatCounter, a leading web traffic and usage analysis tool. The third factor is the seasonal effect, which is commonly observed in time-series data. We control for seasonal effects by using four dummy variables (*Sea1*-*Sea4*) in the model. Table 1 shows the variable definitions and summary statistics.

—— Insert Table 1 here ——

3.3 Model Setup and Estimation

In constructing our empirical model, we use a persistence modeling technique, i.e., the *vector autoregression with exogenous variable* (VARX) (Dekimpe and Hanssens 1995; Luo 2009; Pauwels 2004). This modeling approach has been employed in the recent IS and marketing literature (e.g., Adomavicius et al. 2012; Bang et al. 2013; Luo et al. 2013; Trusov et al. 2009) and it allows us to capture dynamic interactions and feedback effects between the two sides of the platform. In our research context, VARX has several advantages over alternative modeling techniques. First, it can assess both the short-term (i.e., immediate but potentially transient) effects and the long-term (i.e., potentially lagged but stable) effects of explanatory variables on dependent variables. It therefore allows us to compare the duration of application-to-user effect with the user-to-application effect, and explore their asymmetric influences (Chang and Gurbaxani 2012; Luo et al. 2013). Second, VARX models can simultaneously capture the dynamic and intricate mutual influences between different variables, helping better illustrate CNEs between the two sides of the platform.

For the analysis at the aggregated level, we adopt a standard VARX procedure as in Dekimpe and Hanssens (1999). First, we determine the appropriateness of VARX based on Granger causality tests (Granger 1969). Second, we determine the model specification (VARX in levels, VARX in differences, or error-correction forms) based on the unit-root and cointegration test results. Third, using information criteria, we determine the model specification (number of lags). Finally, we derive *generalized impulse response functions* (GIRFs) to assess both short-term and long-term effects.

In line with Trusov et al. (2009) and Adomavicius et al. (2012), we conduct a series of Granger causality tests, to explore whether explanatory variables explain the variation of dependent variables, in addition to the lagged values of dependent variables. Granger causality tests offer valid arguments on causality in time-series data. By using lags of up to 20 periods (Trusov et al. 2009), we investigate whether we need to model a full dynamic system. We find Granger causal relationships between multiple pairs of variables, e.g., the diversity of existing applications Granger causes the user usage (p = .00). These results suggest the need to use a full dynamic model, such as a VARX model.

Model Specification. We use unit root tests to determine whether variables are evolving or stationary. Stationarity of endogenous variables implies that the fluctuations of these variables caused by any unexpected changes will eventually dissipate and these variables will revert back to their deterministic pattern without a permanent regime shift. The variance of stationary variables is finite and time invariant (Dekimpe and Hanssens 1999). We use an augmented Dickey-Fuller test (ADF) to check stationarity. After the first differences, the ADF tests of almost all variables are less than the critical value and can reject the null hypothesis of a unit root (see Table 2). We thus estimate the VARX model using the first differences. Accordingly, we need to determine the appropriate number of lags used for endogenous variables. Based on the Akaike information criterion (AIC) and Schwarz-Bayesian information criterion (SBIC), we select the lag of two periods for analysis (AIC = -21.93, SBIC = -19.33).

—— Insert Table 2 here ——

We first propose a VARX system to capture dynamic interactions between the user side and the application side. We also include a vector of the exogenous variables, an intercept C, and a deterministic-trend variable T that captures the impact of the omitted, gradually changing trend of the dependent variables. The VARX specification is given in Equation 1:

$$\begin{bmatrix} NA_t \\ DA_t \\ UU_t \end{bmatrix} = \begin{bmatrix} C_{NA} \\ C_{DA} \\ C_{UU} \end{bmatrix} + \begin{bmatrix} \delta_{NA} \\ \delta_{DA} \\ \delta_{UU} \end{bmatrix} \times \mathrm{T} + \sum_{j=1}^{J} \begin{bmatrix} \varphi_{11}^{j}\varphi_{12}^{j}\varphi_{13}^{j} \\ \varphi_{21}^{j}\varphi_{22}^{j}\varphi_{23}^{j} \\ \varphi_{31}^{j}\varphi_{32}^{j}\varphi_{33}^{j} \end{bmatrix} \begin{bmatrix} NA_{t-j} \\ DA_{t-j} \\ UU_{t-j} \end{bmatrix} + \sum_{j=1}^{J} \begin{bmatrix} \tau_{11}^{j}\tau_{12}^{j}\tau_{13}^{j} \\ \tau_{21}^{j}\tau_{22}^{j}\tau_{23}^{j} \\ \tau_{31}^{j}\tau_{32}^{j}\tau_{33}^{j} \end{bmatrix} \begin{bmatrix} CE_{t-j} \\ PMS_{t-j} \\ SEA_{t-j} \end{bmatrix} + \begin{bmatrix} \varepsilon_{NA,t} \\ \varepsilon_{DA,t} \\ \varepsilon_{UU,t} \end{bmatrix} \quad (1)$$

In Eq. (1). t is the index of week, J is the maximum number of lags, and ε is a vector of white-noise disturbances with a normal distribution of N(0, Σ). Both δ and φ are parameters of interest. We also test the existence of serial correlation (using a LM test) and heteroskedasticity (using a White test). The significant results from these tests (LM test = 101.14; White test = 7374.90; $p<0.01$) suggest that the model error terms are associated with white noise and the estimated model parameters are unbiased (Joshi and Hanssens 2010).

Next, as VARX model coefficients are not interpretable by themselves (Sims 1980), we use the estimated parameters of the full VARX model to derive the *impulse response functions* (IRFs) and use them to examine the effects of endogenous variables (NA, DA and UU). The impulse response functions trace the over-time impact of one unit shock of an endogenous variable on other endogenous variables. We adopt the *generalized impulse response functions* (GIRFs) because they can ensure that the order of variables in the system does not affect the results, and take into account the contemporaneous effects (Dekimpe and Hanssens 1999; Luo et al. 2013). Standard errors are derived by simulating the fitted VARX model using a Monte Carlo simulation with 1000 runs to test the significance of parameters. Table 3 reports the short-term and long-term effects of endogenous variables on corresponding dependent variables. We follow the tradition in previous VARX research (e.g., Dekimpe and Hanssens 1999; Pauwels and Weiss 2008) to report values of significant effects and leave nonsignificant effects as zero. Figure 1 shows the accumulated impulse responses for both user-to-application and application-to-user effects.

—— Insert Table 3 here ——

—— Insert Figure 1 here ——

To reveal how CNEs are influenced by application review time (AR), we create interaction terms between AR and endogenous variables as follows.

$$\begin{bmatrix} NA_t \\ DA_t \\ NA_t * AR \\ DA_t * AR \\ UU_t \\ UU_t * AR \end{bmatrix} = \begin{bmatrix} C_{NA} \\ C_{DA} \\ C_{NA*AR} \\ C_{DA*AR} \\ C_{UU} \\ C_{UU*AR} \end{bmatrix} + \begin{bmatrix} \delta_{NA} \\ \delta_{DA} \\ \delta_{NA*AR} \\ \delta_{DA*AR} \\ \delta_{UU} \\ \delta_{UU*AR} \end{bmatrix} \times \mathrm{T} + \sum_{j=1}^{J} \begin{bmatrix} \varphi_{1,1}^{j} \cdots \varphi_{1,6}^{j} \\ \varphi_{2,1}^{j} \cdots \varphi_{2,6}^{j} \\ \varphi_{3,1}^{j} \cdots \varphi_{3,6}^{j} \\ \varphi_{4,1}^{j} \cdots \varphi_{4,6}^{j} \\ \varphi_{5,1}^{j} \cdots \varphi_{5,6}^{j} \\ \varphi_{6,1}^{j} \cdots \varphi_{6,6}^{j} \end{bmatrix} \begin{bmatrix} NA_{t-j} \\ DA_{t-j} \\ NA_{t-j} * AR \\ DA_{t-j} * AR \\ UU_{t-j} \\ UU_{t-j} * AR \end{bmatrix} + \sum_{j=1}^{J} \begin{bmatrix} \gamma_{1,1}^{j} \cdots \gamma_{1,6}^{j} \\ \gamma_{2,1}^{j} \cdots \gamma_{2,6}^{j} \\ \gamma_{3,1}^{j} \cdots \gamma_{3,6}^{j} \\ \gamma_{4,1}^{j} \cdots \gamma_{4,6}^{j} \\ \gamma_{5,1}^{j} \cdots \gamma_{5,6}^{j} \\ \gamma_{6,1}^{j} \cdots \gamma_{6,6}^{j} \end{bmatrix} \times \begin{bmatrix} AR_{t-j} \\ AR_{t-j} \\ 0 \\ 0 \\ AR_{t-j} \\ 0 \end{bmatrix} +$$

$$\sum_{j=1}^{J}\begin{bmatrix}\tau_{1,1}^{j}\tau_{1,2}^{j}\tau_{1,3}^{j}\\ \tau_{2,1}^{j}\tau_{2,2}^{j}\tau_{2,3}^{j}\\ \tau_{3,1}^{j}\tau_{3,2}^{j}\tau_{3,3}^{j}\\ \tau_{4,1}^{j}\tau_{4,2}^{j}\tau_{4,3}^{j}\\ \tau_{5,1}^{j}\tau_{5,2}^{j}\tau_{5,3}^{j}\\ \tau_{6,1}^{j}\tau_{6,2}^{j}\tau_{6,3}^{j}\end{bmatrix}\begin{bmatrix}CE_{t-j}\\ PMS_{t-j}\\ SEA_{t-j}\end{bmatrix} + \begin{bmatrix}e_{NA}\\ e_{DA}\\ e_{NA*AR}\\ e_{DA*AR}\\ e_{UU}\\ e_{UU*AR}\end{bmatrix} \tag{2}$$

Table 4 reports the significant short-term and long-term effects of endogenous variables assessed using GIRFs derived from the estimated parameters from the VARX model (2).

—— Insert Table 4 here ——

To examine the moderating effect of platform updates, we follow Pauwels and Weiss (2008) to illustrate the impact of platform updating frequency using two variables: the pulse variable "Move" and the step variable "Update". The pulse variable "Move" is equal to 1 at the time of the move from 0 to 1 with frequent update, and measures direct effects on user and application sides. The step variable "Update" (0 stands for period before frequent updates, 1 stands for period after frequent updates) interacts with both user and application sides, and reveals how their interdependence differs before and after the frequent platform updates. Both variables may have short-term as well as long-term effects. The model specification is as follows:

$$\begin{bmatrix}NA_t\\ DA_t\\ NA_t*Update\\ DA_t*Update\\ UU_t\\ UU_t*Update\end{bmatrix} = \begin{bmatrix}C_{NA}\\ C_{DA}\\ C_{NA*Update}\\ C_{DA*Update}\\ C_{UU}\\ C_{UU*Update}\end{bmatrix} + \begin{bmatrix}\delta_{NA}\\ \delta_{DA}\\ \delta_{NA*Update}\\ \delta_{DA*Update}\\ \delta_{UU}\\ \delta_{UU*Update}\end{bmatrix}\times \mathrm{T} + \sum_{j=1}^{J}\begin{bmatrix}\varphi_{1,1}^{j}\cdots\varphi_{1,6}^{j}\\ \varphi_{2,1}^{j}\cdots\varphi_{2,6}^{j}\\ \varphi_{3,1}^{j}\cdots\varphi_{3,6}^{j}\\ \varphi_{4,1}^{j}\cdots\varphi_{4,6}^{j}\\ \varphi_{5,1}^{j}\cdots\varphi_{5,6}^{j}\\ \varphi_{6,1}^{j}\cdots\varphi_{6,6}^{j}\end{bmatrix}\begin{bmatrix}NA_{t-j}\\ DA_{t-j}\\ NA_{t-j}*Update\\ DA_{t-j}*Update\\ UU_{t-j}\\ UU_{t-j}*Update\end{bmatrix} + \sum_{j=1}^{J}\begin{bmatrix}\gamma_{1,1}^{j}\cdots\gamma_{1,6}^{j}\\ \gamma_{2,1}^{j}\cdots\gamma_{2,6}^{j}\\ \gamma_{3,1}^{j}\cdots\gamma_{3,6}^{j}\\ \gamma_{4,1}^{j}\cdots\gamma_{4,6}^{j}\\ \gamma_{5,1}^{j}\cdots\gamma_{5,6}^{j}\\ \gamma_{6,1}^{j}\cdots\gamma_{6,6}^{j}\end{bmatrix}$$

$$\times\begin{bmatrix}Move_{t-j}\\ Move_{t-j}\\ 0\\ 0\\ Move_{t-j}\\ 0\end{bmatrix} + \sum_{j=1}^{J}\begin{bmatrix}\tau_{1,1}^{j}\tau_{1,2}^{j}\tau_{1,3}^{j}\\ \tau_{2,1}^{j}\tau_{2,2}^{j}\tau_{2,3}^{j}\\ \tau_{3,1}^{j}\tau_{3,2}^{j}\tau_{3,3}^{j}\\ \tau_{4,1}^{j}\tau_{4,2}^{j}\tau_{4,3}^{j}\\ \tau_{5,1}^{j}\tau_{5,2}^{j}\tau_{5,3}^{j}\\ \tau_{6,1}^{j}\tau_{6,2}^{j}\tau_{6,3}^{j}\end{bmatrix}\begin{bmatrix}CE_{t-j}\\ PMS_{t-j}\\ SEA_{t-j}\end{bmatrix} + \begin{bmatrix}e_{NA}\\ e_{DA}\\ e_{NA*Update}\\ e_{DA*Update}\\ e_{UU}\\ e_{UU*Update}\end{bmatrix} \tag{3}$$

Table 5 reports the significant short-term and long-term effects of endogenous variables assessed using GIRFs derived from the estimated parameters from the VARX model (3).

—— Insert Table 5 here ——

4. RESULTS

4.1 Hypotheses Testing

We first assess the short-term and long-term effects of application number and diversity on user usage (i.e., the application-to-user effect), and the short-term and long-term effects of user usage on application number and diversity (i.e., the user-to-application effect). As Table 3 and Figure 1 indicate, user usage has significant long-term effects on both the number and the diversity of existing applications. The short-term effects of user usage on the application-side are not significant. These results suggest that when the installed base of end users expands, the application side does not respond quickly in growth. Instead, the application-side grows slowly but this growth tends to be stable and lasting. In contrast, Table 3 and Figure 1 also show that both the number and the diversity of existing applications have a significant short-term effect on user usage. However, their long-term effects on user usage are not significant. These results indicate that when the installed base of third-party applications grows, the user-side tends to respond with an instant increase in user usage. However, such increase in usage is likely to be temporary and lessen quickly. Therefore, H1 is supported.

With regard to the moderating effect of application review time, Table 4 reveals that application review time significantly moderates the user-to-application effect. Specifically, the interaction between applications review time and user usage has a negative long-term effect on both application number and application diversity. These results suggest that when the average time for application review is longer, the growth of both application number and application diversity as a response to the user-side become weaker in the long-term (i.e., less stable and lasting). Given that the user-to-application effects are primarily long-term effects, H2a is supported. In addition, Table 4 also shows that longer application review time even weakens the short-term effect of user usage on application number, suggesting that longer application review time makes application number respond even less immediately in the short-run to the user-side. However, application review time does not influence the effect of user usage in the short-term on application diversity.

When it comes to the application-to-user effect, Table 4 shows that the interactions between application review time and both application number and application diversity do not significantly influence user usage in either short-term or long-term. In other words, applications review time does not moderate the application-to-user effect at all. Therefore, H2b is not supported. These findings suggest that the delay caused by application review does not significantly undermine the value of applications to end users. A potential explanation is that careful application review, which takes long time, may essentially help ensure application quality and benefit the user-side. We conduct an additional analysis (as explained below) to verify explanation.

Table 5 shows how platform updating frequency moderates the user-to-application effect and application-to-user effect. Specifically, the interaction between platform updating frequency and user usage has a negative long-term effect on both application number and application diversity. The implication is that when the platform updates itself more frequently, the growth of both application number and application diversity as a response to the user-side become weaker in the long-term (i.e., less stable and lasting). In contrast, platform updating frequency has no significant influence on the user-to-application effect in the short-term. Given that the user-to-application effects are primarily long-term effects, H3a is supported.

Table 5 also shows that the interaction between the platform updating frequency and application number has a negative effect on user usage in the short-term. In other words, when the platform updates itself more frequently, it becomes less likely for the user-side to respond to the increased application number with even a short-term temporary growth of usage. However, platform updating frequency does not significantly weaken the response of user usage to application diversity in the short-term. Therefore, H3b is only partially supported. A potential explanation is that the growth in application diversity may provide more complementarity between applications that generate additional value to end user. Such complementarity-based value may also counteract the hassles of platform updating to end users and help attract users to respond to the application-side in the short-term.

4.2 Robustness Check

We conduct several additional analyses to verify the robustness of our results. First, following Pauwels and Weiss (2008), we evaluate the parameter stability of VARX models using a bootstrapping approach. We first construct a new dataset by random re-sampling 100 times from our existing observed dataset. Then, we rerun the VARX models with this bootstrapped dataset. The comparison of the estimates in our main analysis with their corresponding estimates in the bootstrapped dataset suggests that all significant results are consistent. That is, for each significant effect in our main analysis, the value 0 falls out of the 95% confidence interval of the corresponding bootstrap estimate.

Second, our main analysis uses user usage as the measure to characterize the user-side. To further verify the insights of our analysis, we also collect data on user downloading (UD) and use it as an alternative measure to characterize the user-side. We consider the total times of platform downloading and log-transform this variable to address the distribution skewness. We rerun the VARX models with the measure of user usage replaced with UD. The results based on user downloading, as summarized in Appendix A, are qualitatively consistent with the main results based on user usage. Specifically, the increase in user downloading drives long-term growth of both application number and application diversity (in Table A1), with this long-term effect weakened by both long application review time (in Table A2) and frequent platform updating (in Table A3). The increase in application number drives the short-term growth of user downloading (in Table A1), with this short-term effect weakened by both long application review time (in Table A2) and frequent platform updating (in Table A3). However, application diversity has no significant effect on user downloading. A potential explanation is that as downloading is costless, users are more likely to download and try different applications without necessarily committing to them in usage. Therefore, the growth only in application diversity but not in total application volume does not necessarily trigger any significant increase in user downloading.

Third, to further verify the moderating effect of platform updating frequency, we also split the sample and estimate the model for both the time period before the change of platform updating policies and the time period after the policy change. We examine the difference between these two periods in terms of the short-term and long-term effects of both sides. The results are consistent with those of the main analysis. Specifically, before the change of platform updating policies (i.e., when the platform was updated less frequently), user usage has a significant long-term effect on both application number and application diversity. After the change of platform updating policies (i.e., when the platform was updated more frequently), in contrast, neither the short-term effect nor the long-term effect of user usage is significant. Similarly, before the policy change, application number has a significant short-term effect on user usage. After the the policy change, this effect is insignificant. Application diversity, however, has a short-term effect on user usage in both the pre-change period and the post-change period. In this regard, platform updating frequency weakens the user-to-application effect and partially weakens the application-to-user effect.

4.3 Additional Insights

We conduct some additional analyses to develop more insights that complement our main findings. As mentioned above, we find that the application-to-user effect is not significantly weakened by application review time. A potential explanation is that careful application review, although time-consuming, may essentially help improve application quality and counteract the negative effect of delay in application release. We therefore conduct an additional analysis to see if this explanation is plausible. We collect additional weekly data on the quality of new applications. In the official website for Firefox applications, end users can rate downloaded applications after they use them. The rating is based on a 5-star scale, where one star means *extremely low quality* and five stars mean *extremely high quality*. For each new application, we used the average user rating as a measure of quality. We then create a separate VARX system including length of applications review, quality of new applications, and other exogenous variables. The results, as summarized in Table B1 of Appendix B, show that long application time significantly increases quality of new applications in both the short-term and the long-term. This finding provides a key insight about why application review time may not necessarily weaken the application-to-user effect. That is, the application quality improvement caused by long application review is likely to offset the negative impact of application delay. It is also worth noticing that, however, such application quality improvement is still not enough to eventually result in any long-term increase of user usage, potentially due to the intense platform competition in the online world.

Second, we have mentioned above that a potential reason for the negative moderating effect of platform updating frequency on the user-to-application effect is that application developers may be forced to devote more resources to updating their existing applications, making it more challenging for them to develop new applications to respond to the user-side. We conduct an additional analysis to develop more insights. Specifically, we specify a new VARX model incorporating frequent platform updates, number of updated applications, number of new applications, and other exogenous variables. The results, as summarized in Table B2 of Appendix B, show that more frequent platform updating significantly increases the number of updated applications in both the short-term and the long-term. In addition, the number of updated applications significantly decreases number of new applications in both the short-term and the long-term. Therefore, these findings support the explanation that frequent platform updating weakens the user-to-application effect by limiting developers' efforts in developing more new applications as a response to the user-side.

5. DISCUSSION

5.1 Theoretical Implications

This study provides several important findings that contribute to multiple streams of literature. First, this study complements the existing literature on platform business and two-sided markets. Different from other market environments, two-side markets are characterized by CNEs through which the two sides of the platform influence each other (Parker and Van Alstyne 2005; Rochet and Tirole 2006). The existing literature on two-sided markets has mostly focused on the platform's strategies on different sides (e.g. Eisenmann et al. 2006; Parker and Van Alstyne 2008; Seamans and Zhu 2014). In prior studies, however, CNEs are assumed to remain temporally stable and symmetric between the application and user sides. Our study theoretically explains and empirically illuminates how CNEs in different directions—application-to-user and user-to-application—may be asymmetric in immediacy and duration. Specifically, we find a long-term CNE from the user side to the application side and a short-term CNE from the application side to the user side. These results illustrate key differences between the application and user sides in their value creation and value capture processes, and also help better understand the platform's performance in interacting with both sides.

Second, while prior research focuses on the case where the platform takes CNEs as given, this study investigates how platform governance mechanisms influence CNEs between the two sides. A general finding is that CNEs are likely to be weakened by various platform governance policies, such as comprehensive review processes and frequent updating of the platform itself. The implication is that platform governance can influence how the application and user sides co-evolve by engaging in monitoring and quality assurance (comprehensive review processes) and adaptation of the platform (frequent updating). This finding about the role of IS governance in attenuating CNEs also contributes to both the research on software versioning strategies and the general research stream on IS governance. While the existing research on software versioning focuses more on the impact of upgrading on the diffusion and market performance of focal software (e.g., Ji et al. 2005), our analysis sheds light on the more profound impacts of platform updating on the ecosystem encompassing other complementary applications, end users, and their mutual influences. In addition, our analysis extends the dominant perspective that has focused on IS governance as an internal organizational arrangement (e.g., Sambamurthy and Zmud 1999) or an inter-organizational arrangement of dyadic relationships and a firm's relationship portfolios (Klein and Rai 2009; Rai and Tang 2010) to considering specific IS governance mechanisms in the new platform context (Tiwana et al. 2010).

Third, this study also adds to the emerging and growing IS literature that considers the complex and adaptive features of information systems in a digitally networked context (e.g., Tanriverdi et al.

2010). This literature provides theoretical perspectives to recognize and characterize the evolving nature of competitive landscape and the need to dynamically reconfigure IS strategies to achieve a series of temporary advantages over time. In the context of software platform ecosystem, our study considers that the user and application sides of a market change in composition and diversity over time and, importantly, reveals that these sides co-evolve through CNEs from each side to the other. What is especially important from a theoretical perspective is that underlying the co-evolution are asymmetric influence mechanisms in immediacy and duration between the two sides. Our study also suggests that in CABS the trajectory and direction of co-evolution of parties can be affected by the actor in a central governance role, in our case the platform. The central actor in a governance role can establish monitoring and quality control rules that moderate the nature of the interdependence among parties, as we found with cumulative reviews mitigating the CNEs. It needs to also have the adaptive capacity to effectively moderate these interdependencies, as was revealed by our findings of the role of frequent platform updating as an attenuator of the CNEs. The implications are that interdependencies among diverse parties in CABS are not only likely to be asymmetric in immediacy and duration but that these dynamics can be affected by a central actor through the enactment of digital governance mechanisms.

5.2 Practical Implications

Our study also has important practical implications. First, the characterization of the asymmetry in CNEs has implications for how platform managers can effectively coordinate the ecosystem. Platforms with two-sided markets often face a challenge of coordinating the interactions of buyers and sellers, especially as CNEs enable participants on each side to form their expectations about the other (Spulber 2010). To solve the coordination problem, platforms need to decide which side to prioritize in resource commitment (Hagiu and Spulber 2013). Our finding on the differences in immediacy and duration of the user-to-application and the application-to-user effects can inform this decision. The long-term user-to-application effect suggests that when the platform tends to leverage its installed base on the user side to enrich its application side, the platform owners need to be more patient and persistent in resource commitment. However, the platform may expect stable benefits in the long-run from leveraging the user-to-application effect. In contrast, when the platform tends to leverage its application base to attract more users, the platform owners needs to be more cautious about the unsustainability of the usage growth resulting from the application-to-user effect, although it may also benefit temporarily from this effect.

Second, our study highlights the needs for platform managers to actively monitor and manage CNEs. Many platform governance strategies such as application review, while adopted to improve product offerings to users, may generate unexpected impact on CNEs and the ecosystem. Our study reveals that although application review improves application quality, it may also weaken the positive network effect of the user side on the growth of application side. When developers have to go through long review processes to eventually release their applications, the response of the application side to the increased installed base of platform users become both less immediate in the short-term and less durable in the long-term. Long review processes also keep the increased user base from stimulating more complementary applications in the long-run. In this regard, platform managers should be serious about improving the efficiency of application review processes to not adversely impact positive CNEs.

Third, the findings on the impact of platform updating frequency also help to call for platform managers' attention to the consequences of platform updating strategies. Our study suggests that platform updating affects both the user and application sides, and can significantly undermine the mutual influences between these two sides. Frequent updating of the platform, although possibly beneficial to the ecosystem from the standpoint of enhanced functionalities and efficiency (Khoo and Robey 2007), limits the capabilities of both sides to benefit from the growth of the other. Platform managers, therefore, should figure out strategies to insulate or restore CNEs when updating the platform. The asymmetric nature of CNEs also suggests that platform strategies to aid the CNE from each side may require different focuses. The effort to maintain the user-to-application effect may require a focus on encouraging new application development and supporting the long-term sustainable reaction of the application side in growth. In contrast, the effort to maintain the application-to-user effect may require a focus on facilitating users to more instantly access the application resources as well as preventing the user interests from fading out quickly.

6. CONCULSION

This study investigates the asymmetric feature of CNEs in the software platform ecosystem and how platform governance mechanisms may influence CNEs. Our findings suggest that the application-to-user effect is primarily short-term and the user-to-application effect is primarily long-term. Long application review time significantly weakens the user-to-application effect, but not the application-to-user effect. Our additional evidence suggests that long application review time results in enhanced quality of applications that may counteract the negative impact of delayed applications (caused by long application time) to the user side. In addition, we find that frequent platform updating weakens the user-to-application effect, in part because frequent platform updating forces developers to commit more efforts to updating existing applications rather than developing new applications. Frequent platform updating also partially weakens the application-to-user effect by mitigating the attractiveness of increased application numbers, but not increased application diversity, to end users.

This study also has some limitations that leave room for future research. First, our field data is about a specific web browser platform. Future research may consider verifying the key insights about CNEs from this study in other IT-based platform model contexts (e.g., Web services, open-source systems, mobile OS, etc). The influences of other platform-specific technological and non-technological factors on CNEs are also worth exploring. Second, this study focuses on the case of free platform software and third-party applications. As a result, the role of pricing is not relevant to the context of our study. However, pricing is an important control mechanism of the platform on both the application and user sides (Parker and Van Alstyne 2008), and is worth further exploring in future work. In addition, the examination of pricing issues can help shed light on how CNEs may influence the competition between the platform and third-party application developers for profits. Therefore, future research may focus on platforms with pricing and specifically consider the issue of profitability. Fourth, while our study focuses primarily on the interactions between a platform and its own two-sided markets, future research may explore how the evolving competition between different platforms may be influenced by asymmetric CNEs and the interactions between CNEs and platform strategies. Future research can identify and evaluate how platform strategies strengthen, rather than weaken, CNEs and enable platforms to accrue competitive advantages over others. Some situations in which platforms may intentionally weaken CNEs so as to seek more controls over the two sides also deserve more research attention.

REFERENCES

Adomavicius, G., Bockstedt, J., and Gupta, A. 2012. “Modeling Supply-Side Dynamics of IT Components, Products, and Infrastructures: An Empirical Analysis Using Vector

Autoregression," *Information Systems Research* (23:2), pp. 397-417.
Afuah, A. 2013. "Are Network Effects Really All about Size? The Role of Structure and Conduct," *Strategic Management Journal* (34:3), pp. 257-273.
Allen, P. M., and Varga, L. 2006. "A Co-Evolutionary Complex Systems Perspective on Information Systems," *Journal of Information Technology* (21:4), pp. 229–238.
Anderson, E.G., Parker, G.G., and Tan, B. 2014. "Platform Performance Investment in the Presence of Network Externalities," *Information Systems Research* (25:1), pp. 152-172.
Bang, Y., Lee, D.J., Han, K., Hwang, M., and Ahn, J.H. 2013. "Channel Capabilities, Product Characteristics, and the Impacts of Mobile Channel Introduction," *Journal of Management Information Systems* (30:2), pp. 101-125.
Basu, A., Mazumdar, T., and Raj, S.P. 2003. "Indirect Network Externality Effects on Product Attributes," *Marketing Science* (22:2), pp. 209-221.
Benbya, H., and McKelvey, B. 2006. "Using Coevolutionary and Complexity Theories to Improve IS Alignment: A Multi-Level Approach," *Journal of Information Technology* (21:4), pp. 284-298.
Boudreau, K.J. 2010. "Open Platform Strategies and Innovation: Granting Access vs. Devolving Control," *Management Science* (56:10), pp. 1849-1872.
Boudreau, K.J. 2012. "Let a Thousand Flowers Bloom? An Early Look at Large Numbers of Software App Developers and Patterns of Innovation," *Organization Science* (23:5), pp. 1409-1427.
Bowman, C., and Ambrosini, V. 2000. "Value Creation Versus Value Capture: Towards a Coherent Definition of Value in Strategy," *British Journal of Management* (11:1), pp. 1-15.
Ceccagnoli, M., Forman, C., Huang, P., and Wu, D.J. 2012. "Cocreation of Value in a Platform Ecosystem: The Case of Enterprise Software," *MIS Quarterly* (36:1), pp. 263-290.
Chang, Y.B. and Gurbaxani, V. 2012. "The Impact of IT-Related Spillerovers on Long-Run Productivity: An Empirical Analysis," *Information Systems Research* (23:3), pp. 868-886.
Chatain, O., and Zemsky, P. 2011. "Value Creation and Value Capture with Frictions," *Strategic Management Journal* (32:11), pp. 1206-1231.
Cusumano, M., and Gawer, A. 2002. "The Elements of Platform Leadership," *Sloan Management Review* (43:3), pp. 51-58.
Dekimpe, M. G. and Hanssens, D. M. 1995. "Empirical Generalizations about Marketing Evolution and Stationarity," *Marketing Science* (14:3), pp. 109-121.
Dekimpe, M. G. and Hanssens, D. M. 1999. "Sustained Spending and Persistent Response: A New Look at Long-Term Marketing Profitability," *Journal of Marketing Research* (36:4), pp. 397-412.
Eisenmann, T.R., Parker, G., and Van Alstyne, M. 2006. "Strategies for Two Sided Markets," *Harvard Business Review* (84:10), pp. 92–101.
Eisenmann, T.R., Parker, G., and Van Alstyne, M.W. 2010. "Opening Platforms: How, When and Why?," in *Platforms, Markets and Innovation*, A. Gawer (ed.), Northampton, MA: Edward Elgar, pp. 131-162.
Eisenmann, T.R., Parker, G., and Van Alstyne, M.W. 2011. "Platform Envelopment," *Strategic Management Journal*, (32:12), pp. 1270-1285.
Evans, D.S., Hagiu, A., and Schmalensee, R. 2006. *Invisible Engines: How Software Platforms Drive Innovation and Transform Industries*. Cambridge, MA: MIT Press.
Ghazawneh, A., and Henfridsson, O. 2013. "Balancing Platform Control and External Contribution in Third-Party Development: The Boundary Resource Model," *Information Systems Journal* (23:2), pp. 173-192.

Granger, C.W.J. 1969. “Investing Causal Relations by Econometric Models and Cross-Spectral Methods,” *Econometrica* (37:3), pp. 424-438.

Hagiu, A. 2006. “Pricing and Commitment by Two-Sided Platforms,” *RAND Journal of Economics* (37:3), pp.720–737.

Hagiu, A., and Spulber, D. 2013. “First-Party Content and Coordination in Two-Sided Markets,” *Management Science* (59:4), pp. 933-949.

Hilkert, D., Benlian, A., and Hess, T. 2010. “Motivational Drivers to Develop for Social Software-Platforms: The Example of Facebook,” *AMCIS 2010 Proceedings*, Paper 86.

Ji, Y., Mookerjee, V.S., and Sethi, S.P. 2005. “Optimal Software Development: A Control Theoretic Approach,” *Information Systems Research* (16:3), pp. 292-306.

Joshi, A. and Hanssens, D. M. (2010). “The Direct and Indirect Effects of Advertising Spending on Firm Value,” *Journal of Marketing* (74:1), pp. 20-33.

Khoo, H. M., and Robey, D. 2007. “Deciding to Upgrade Packaged Software: A Comparative Case of Motives, Contingencies and Dependencies,” *European Journal of Information Systems* (16:5), pp. 555-567.

Kim, B.C. Lee, J.J. and Park, H. 2013. “Platform Entry Strategy in Two-Sided Markets: Evidence from the Online Daily Deals Industry,” working paper, Georgia Institute of Technology.

Klein, R., and Rai, A. 2009. “Interfirm Strategic Information Flows in Logistics Supply Chain Relationships,” *MIS Quarterly* (33:4) pp. 735-762

Lepak, D.P., Smith, K.G., and Taylor, M.S. 2007. “Value Creation and Value Capture: A Multilevel Perspective,” *Academy of Management Review* (32:1), pp. 180-194.

Lieberman, M.B. 2007. “Did First-Mover Advantage Survive the Dot-Com Crash?” working paper, UCLA Anderson School of Management, Los Angeles, CA.

Luo, X. 2009. “Quantifying the Long-Term Impact of Negative Word of Mouth on Cash Flows and Stock Prices,” *Marketing Science* (28:1), pp. 148-165.

Luo, X., Zhang, J., and Duan, W. 2013. “Social Media and Firm Equity Value,” *Information Systems Research* (24:1), pp. 146-163.

Maurer, C. and Tiwana, A. 2012. “Control in App Platforms: The Integration-Differentiation Paradox,” *ICIS 2012 Proceedings*.

McKelvey, B. 1997. “Quasi-natural Organization Science,” *Organization Science* (8:4), pp. 352-380.

Park, S. 2004. “Quantitative Analysis of Network Externalities in Competing Technologies: The VCR Case,” *Review of Economics and Statistics* (86:4), pp. 937-945.

Parker, G., and Van Alstyne, M.W. 2005. “Two-Sided Network Effects: A Theory of Information Product Design,” *Management Science* (51:10), pp. 1494-1504.

Parker, G. and Van Alstyne, M. W. 2008. “Managing Platform Ecosystems,” *ICIS 2008 Proceedings*, Paper 53.

Pauwels, K. 2004. “How Dynamic Consumer Response, Competitor Response, Company Support, and Company Inertia Shape Long-Term Marketing Effectiveness,” *Marketing Science* (23:4), pp. 598-610.

Pauwels, K. and Weiss, A. 2008. “Moving from Free to Fee: How Online Firms Market to Change their Business Model Successfully,” *Journal of Marketing* (72:3), pp. 14-31.

Rai, A., and Tang, X. 2010. “Leveraging IT Capabilities and Competitive Process Capabilities for the Management of Interorganizational Relationship Portfolios,” *Information Systems Research* (21:3), pp. 516-542.

Rai, A., and Tang, X. 2014. “Information Technology-Enabled Business Models: A Conceptual Framework and a Coevolution Perspective for Future Research,” *Information Systems Research* (25:1), pp. 1-14.

Rochet, J.C. and Tirole, J. 2006. “Two-Sided Markets: A Progress Report,” *RAND Journal of Economics* (37:3), pp. 645–667.

Sambamurthy, V., and Zmud, W. 1999. “Arrangements for Information Technology Governance: A Theory of Multiple Contingencies,” *MIS Quarterly* (23:2), pp. 261-290.

Seamans, R. and Zhu, F. 2014. “Responses to Entry in Multi-Sided Markets: The Impact of Craigslist on Local Newspapers,” *Management Science* (60:2), pp. 476-493.

Sims, C. A. 1980. “Macroeconomics and Reality,” *Econometrica* (48:1), pp. 1-48.

Spulber, D.F. 2010. “Solving the Circular Conundrum: Communication and Coordination in Internet Markets,” *Northwestern University Law Review* (104:2), pp. 537-591.

Starr, M. 2014. “Apple Lists the Top 10 Reasons Apps Get Rejected,” *CNET* (September 2, 2014). Available at: http://www.cnet.com/news/apple-lists-the-top-10-reasons-apps-get-rejected/

Stremersch, S., Tellis, G.J., Franses, P.H., and Binken, J.L.G. 2007. “Indirect Network Effects in New Product Development,” *Journal of Marketing* (71:3), pp. 52-74.

Sun, J. 2012. “Why Different People Prefer Different Systems for Different Tasks: An Activity Perspective in Technology Adoption in a Dynamic User Environment,” *Journal of the American Society for Information Science and Technology* (63:1), pp. 48-63.

Tanriverdi, H., Rai, A., and Venkatraman, N. 2010. “Reframing the Dominant Quests of Information Systems Strategy Research for Complex Adaptive Business Systems,” *Information Systems Research* (21:4), pp. 822-834.

Tiwana, A., Konsynski, B., and Bush, A.A. 2010. “Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics,” *Information Systems Research* (21:4), pp. 675-687.

Trusov, M., Bucklin, R. E., and Pauwels, K. 2009. “Effects of Word-of-Mouth Versus Traditional Marketing: Findings from an Internet Social Networking Site,” *Journal of Marketing* (73:5), pp. 90-102.

Venkatesh, V., Brown, S. A., Maruping. L. M., and Bala, H. 2008. “Predicting Different Conceptualizations of System Use: The Competing Roles of Behavioral Intention, Facilitating Conditions, and Behavioral Expectation,” *MIS Quarterly* (32:3), pp. 483-502.

Weyl, E.G. 2010. “A Price Theory of Multi-Sided Platforms,” *American Economic Review* (100:4), pp. 1642–1672.

Wilbur, K. C. 2008, “A Two-Sided, Empirical Model of Television Advertising and Viewing Markets,” *Marketing Science* (27:3), pp. 356-378.

Zhang, K., Evgeniou, T., Padmanabhan, V. and Richard, E. 2012, “Content Contributor Management and Network Effects in a UGC Environment,” *Marketing Science* (31:3), pp. 433-447.

Zhu, F., and Iansiti, M. 2012. “Entry into Platform-Based Markets,” *Strategic Management Journal* (33:1), pp. 88-106.

Table 1: Constructs, Measures, and Descriptive Statistics					
Variables	**Measurement**	**Mean**	**S.D.**	**Min**	**Max**
Number of apps (NA)	The quantity of existing apps on Firefox each week	8.577	0.449	7.717	9.229
Diversity of apps (DA)	The Hirfindahl index of existing apps on Firefox each week	0.846	0.002	0.841	0.848
User usage (UU)	Weekly amount of actually usage of Firefox by end users	19.385	0.459	18.697	20.016
Applications review (AR)	The average waiting times for reviewing newly launched apps	2.047	0.345	1.609	3.023
Frequent platform update	Dummy variable (=1 after frequent update, =0 before frequent update)	0.601	0.491	0.000	1.000

Table 2: Unit Root Test Result after First Differences		
Variable	**Test Statistic**	***p*-value**
Number of Applications	-5.540	<.0001
Diversity of Applications	-12.653	<.0001
User Usage	-13.868	<.0001
Applications Review	-14.513	<.0001

Table 3: User-to-Application and Application-to-User Effects		
Panel A: User-to-Application Effects		
Paths	**Short-Term Effect**	**Long-Term Effect**
User Usage → App Number	0	0.14
User Usage → App Diversity	0	0.06
Panel B: Application-to-User Effects		
Paths	**Short-Term Effect**	**Long-Term Effect**
App Number → User Usage	0.60	0
App Diversity → User Usage	0.50	0

Notes: All non-zero effects are significant (p<0.05); insignificant effects are displayed as 0. The coefficients are percentage values.

Table 4: Moderating Effects of Applications Review Time on Cross-Side Network Effects		
Panel A: User-to-Application Effects		
Paths	**Short-Term Effect**	**Long-Term Effect**
User Usage × App Review Time → App Number	-0.02	-0.01
User Usage × App Review Time → App Diversity	0	-0.06
Panel B: Application-to-User Effects		
Paths	**Short-Term Effect**	**Long-Term Effect**
App Number × Apps Review Time → User Usage	0	0
App Diversity × Apps Review Time → User Usage	0	0

Notes: All non-zero effects are significant (p<0.05); insignificant effects are displayed as 0. The coefficients are percentage values.

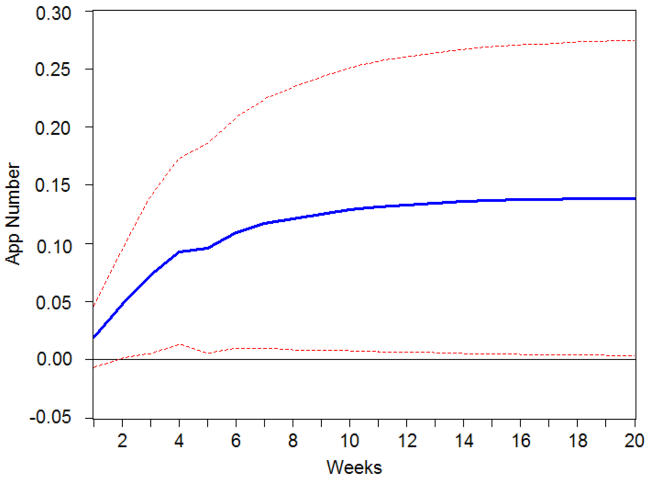
Table 5: Moderating Effects of Platform Updating Frequency on Cross-Side Network Effects		
Panel A: User-to-Application Effects		
Paths	**Short-Term Effect**	**Long-Term Effect**
User Usage × Platform Updating Frequency → App Number	0	-0.12
User Usage × Platform Updating Frequency → App Diversity	0	-0.10
Panel B: Application-to-User Effects		
Paths	**Short-Term Effect**	**Long-Term Effect**
App Number × Platform Updating Frequency → User Usage	-1.19	0
App Diversity × Platform Updating Frequency → User Usage	0	0

Notes: All non-zero effects are significant (*p*<0.05); insignificant effects are displayed as 0. The coefficients are percentage values.

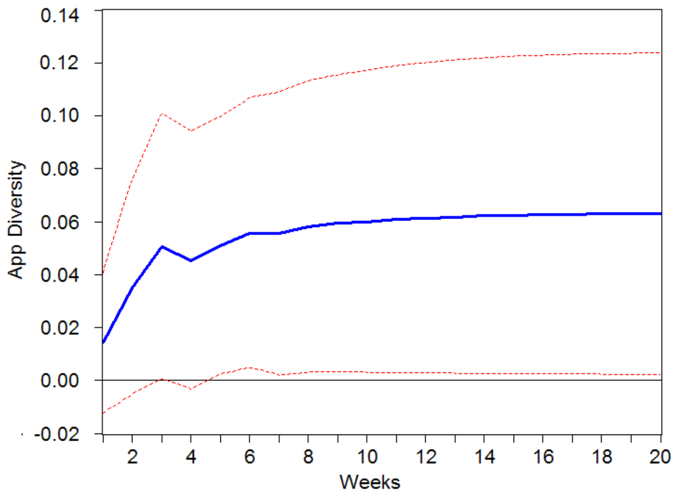
Figure 1: Accumulated Impulse Response Functions of Cross-Side Network Effects

Panel A: User-to-Application Effects

User Usage on App Number

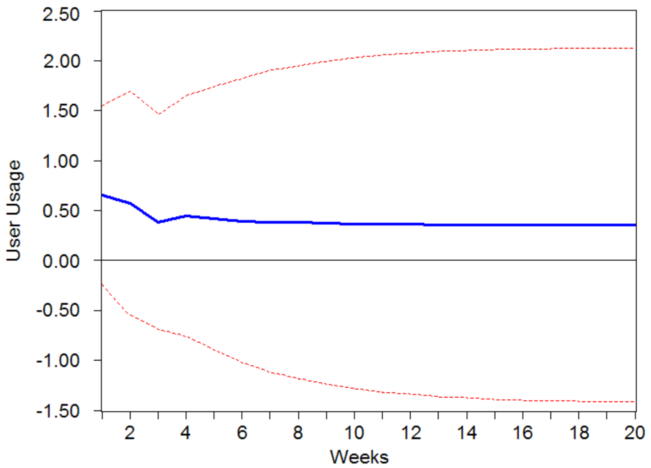


User Usage on App Diversity



Panel B: Application-to-User Effects

App Number on User Usage



App Diversity on User Usage