

User-Generated Open Source Products: Founder's Social Capital and Time to Product Release

Girish Mallapragada

Kelley School of Business, Indiana University, Bloomington, Indiana 47405,
gmallapr@indiana.edu

Rajdeep Grewal, Gary Lilien

Smeal College of Business, Pennsylvania State University, University Park, Pennsylvania 16802
{rgrewal@psu.edu, glilien@psu.edu}

Volunteer users employ collaborative Internet technologies to develop open source products, a form of user-generated content, where time to product release is a crucial measure of project success. The open source community features two separate but related subcommunities: developer users who contribute time and effort to develop products and end users who act as collaborative testers and provide feedback. We develop hypotheses concerning how the location of the project's founders in the social network of developer users, the interplay of developer users and end users, and project and product characteristics affect time to product release. We use data on 817 development projects from SourceForge, a large open source community forum, to calibrate a split hazard model to test the hypotheses. That model supports the two-community conceptualization and most of the related hypotheses. The results have theoretical and managerial implications; for example, a pivotal position of founders in the developer user community can reduce time to product release by up to 31%, and projects in which users are more engaged can experience an 11% shorter time to product release compared with those projects in which they are not.

Key words: user-generated products; product development; social networks; open source; innovation

History: Received: January 24, 2010; accepted: October 3, 2011; Russell Winer served as the special issue editor and Kannan Srinivasan served as associate editor for this article. Published online in *Articles in Advance* February 16, 2012.

1. Introduction

Blogs, Twitter feeds, social network profiles, user videos, and other forms of digital products have the common feature that the associated content's consumers are also its producers; hence such content is known as user-generated content. In the open source software (OSS) domain, where the user-generated content is a software product (Raymond 1999), users are developers of software code as well as consumers of the resulting software products. Because the intellectual property or source code is available in the public domain, these products are referred to as *open source products*. The Android mobile operating system, Firefox browser, Apache Web server, and Linux operating system are a few examples of some successful open source products (O'Mahony and Ferraro 2007).

The OSS paradigm provides an alternative to traditional, top-down, firm-driven software new product development (NPD) processes (e.g., Lakhani and von Hippel 2002, Shah 2006, von Krogh and von Hippel 2006). In OSS development, NPD pro-

cess success depends on the contributions of volunteer users who both produce and use the products (e.g., O'Hern et al. 2011); various NPD projects thus compete for the community's attention and resources (e.g., Subramaniam et al. 2009). When project teams release working product versions quickly, they earn an early mover advantage that offers them the ability to appeal to the community earlier than their competitors and thereby benefit from the community's contributions (e.g., Lakhani and von Hippel 2002, Raymond 1999). Therefore, the time to product release is a critical metric for NPD process success in the OSS domain and the focus of our research.

The user community for an OSS project consists of two subcommunities: developer users, who include OSS project members and developers, and end users, who just download and use the product, though they might develop other projects. Drawing on the NPD literature in both traditional firm settings (for a meta-analysis, see Henard and Szymanski 2001) and the OSS domain (Grewal et al. 2006, Lee et al. 2009),

and recognizing the critical roles both subcommunities play, we propose three factors that influence OSS NPD process success: (1) the information and resources available to the project, which depend on the location of the project founders in the social network of developer users (Burt 1997, Lin 2001); (2) the interplay of the project's developer user and end user communities; and (3) the characteristics of OSS projects and the resulting products.

Project founders' social capital in the network of developer users is the basis for the first factor; we thus draw on literature from sociology pertaining to social networks and social capital (Lin 2001, Obstfeld 2005) and measure social capital through the embeddedness and brokerage activities of founders. Embeddedness measures the volume of information and resource input founders receive from the network; brokerage measures the novelty of such information and resources (Fleming and Waguespack 2007, Nerkar and Paruchuri 2005). For the second factor, we measure both user engagement, which reflects how the end user community participates in the projects (Stewart et al. 2006), and the product's audience focus, which determines which community the products target, i.e., the developer users or end users (Subramaniam et al. 2009). For the third factor, we consider several OSS project and product characteristics, including the project founders' experience, type of open source license, and team size (Stewart et al. 2006, Subramaniam et al. 2009). We model time to product release with a split hazard model to account for projects that never release a product. We collect data over 42 months and include 817 projects that began in January 2003 on SourceForge.net.

Results from a simulated maximum likelihood estimation of the split hazard model show that increases in both embeddedness and brokerage shorten the time to product release of new open source projects. In addition, the engagement of end users and product's audience focus significantly affect the influence of embeddedness on this time. Beyond their statistical significance, our results have managerial implications; for example, time to product release declines by an average of 31% when project founders have both high embeddedness and high brokerage.

In the next section, we discuss the OSS context, develop the central position that time to product release plays in that context, and introduce our conceptual framework. We next develop theoretical arguments for the main effects in our model. Then we outline our timing process model and describe our data source, data collection procedure, and measures. Next we discuss our findings and model validation approach, followed by the implications of our work for theory and practice, its limitations, and opportunities for further research.

2. Conceptual Background

The fundamental principle of the OSS model is simple: OSS community members who develop new digital products (developer users) post the source code on Internet forums (e.g., SourceForge, RForge) under an open source license, and other OSS community members test and use the product (end users), submit feedback, and may modify the code and create their own versions as long as they adhere to the open source licensing restrictions. The term "cocreation" describes such collaboration because content is created and used by the same set of members (O'Hern et al. 2011). In this model, the users' decision to participate in a particular project often depends on the project's visibility, uniqueness, and popularity (Lakhani and von Hippel 2002, Subramaniam et al. 2009). Developer users contribute by writing and modifying code, resolving bugs, maintaining documentation, and addressing support requests; end users contribute by reporting bugs, downloading products, and participating in forums, which generate product visibility. Thus, the creation and maintenance of user-generated content involves the active and iterative interplay of the developer user and end user communities.

Raymond's (1999, p. 29) call to "release early, release often" is a central tenet of the OSS model. A product version that performs only core functions but lacks secondary features or final aesthetics can be released to generate rapid feedback and engagement from the user community. Releasing a product early may provide initial traction and attract dedicated users who offer vital early, critical feedback (Lakhani and von Hippel 2002, Raymond 1999). Furthermore, the first file release is a credible signal that the project team is serious about the project, which fosters interest and engagement from both the developer and user communities. Therefore, time to product release is a critical measure of success for OSS development.¹

Evidence from meta-analytic studies regarding the drivers of traditional NPD suggest that success depends on the product characteristics, organizational processes and strategies, and marketplace characteristics (Henard and Szymanski 2001). Similar OSS research suggests that OSS project and product characteristics, such as type of license, availability of prior code, user and developer interest (e.g., Subramaniam et al. 2009), and network locations of project managers (Grewal et al. 2006), affect open source project

¹ This success metric has limitations (Bayus et al. 1997, Langerak and Hultink 2006), but research also shows that by investing heavily in product research and development (R&D), firms release more frequent product upgrades (Bayus et al. 1997), capture and maintain market share (Smith 1999), and increase future market valuation (Hendricks and Singhal 1997).

success too. Thus, the success of an OSS NPD process likely depends on the information and resources available and the characteristics of the product (Barczak et al. 2009, Lee et al. 2009). Furthermore, because the OSS community consists of two distinct subcommunities—developer users and end users—their interplay should also affect OSS NPD success.

In particular, the availability of information and resources for a project depends on the social capital of developer users working on the project, as reflected in their location in the social network of other developer users (Lin 2001) and their interactions with end users (Shah 2006). As in Figure 1, the drivers of success thus should include (1) the founders' social capital, (2) the interplay of developer users and end users, and (3) control variables related to OSS project and product characteristics. Because developer users drive the NPD process, the effect of their social capital should be moderated by variables concerning the interplay of the developer user and end user subcommunities.

Literature on social capital and social networks (Lin 2001) shows that sufficient embeddedness and brokerage (Burt 2005, Granovetter 1985) provide the necessary resources for NPD projects (Fleming et al. 2007, Obstfeld 2005). When a new project begins, the founders' social network represents the primary connection to the OSS community and a critical determinant of information and resource availability. Research in organizational ecology (Romanelli 1989, Stinchcombe 1965) also shows that initial resource environments have lasting effects on subsequent organizational structures, processes, strategies, and survival likelihood. In the OSS context, the social network of the founders therefore should be the primary

source of information and resources at founding, and its characteristics should influence the success of OSS projects (e.g., Fleming and Waguespack 2007, Grewal et al. 2006).

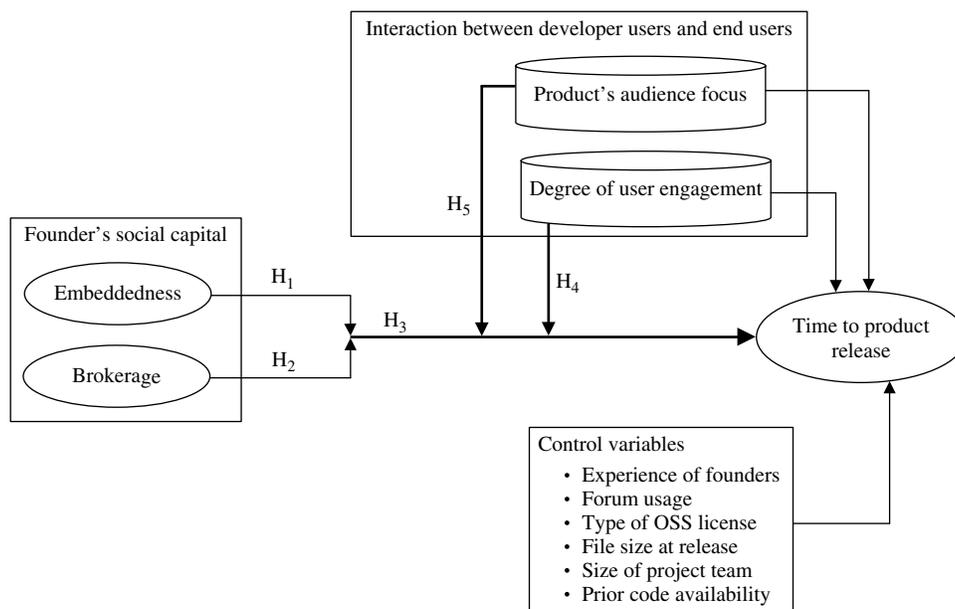
The interplay of developer user and end user communities can be characterized by two variables. First, the degree of user engagement captures the level of interest in the user community and indicates the level of information transfer between communities, forming a channel by which feedback from the end user community reaches the developer user community (Stewart et al. 2006). Second, the product's audience focus represents whether the developer user community presents products to the end user community as "developer-focused" (i.e., technical products) or "end user-focused" (i.e., nontechnical) (Subramaniam et al. 2009). We theorize that the degree of user engagement and product's audience focus both moderate the influence of the developer user social network on time to product release.

Finally, following research on firm-centric NPD processes (Henard and Szymanski 2001), we incorporate the following project and product characteristics as control variables: the type of OSS license, the size of the first release file, the size of the project team, and the availability of prior project code (e.g., Stewart et al. 2006, Subramaniam et al. 2009).

2.1. Founders' Social Capital

Collaboration in goal-oriented groups occurs either through direct, natural collaboration between individual actors who interact (Coleman 1988, Granovetter 1985) or through brokerage (Burt 2005, Fleming and Waguespack 2007). In either case, collaboration

Figure 1 Conceptual Framework



signals trust and entails information and knowledge (resource) sharing (Uzzi 1997). The resource benefits from direct collaboration are higher when the actor is more deeply embedded in the network; higher embeddedness implies greater access to “fine-grained information” that tends to be “tacit, complex, or proprietary” (Fleming et al. 2007, p. 444). Resource benefits from brokered collaboration instead reflect the strength of weak ties (Granovetter 1973), which connect distant parts of a network, so actors that broker weak ties gain access to diverse information and control information sharing (Burt 1997). To capture both forms of collaboration, we characterize founders’ social capital as incorporating both embeddedness and brokerage (see Grewal et al. 2006, Nerkar and Paruchuri 2005). For example, executing innovation requires the mobilization of available resources, which vary directly with embeddedness, but creativity often stems from integrating diverse information, obtained from social capital such as brokerage (Obstfeld 2005).

In Figure 2 we illustrate embeddedness and brokerage in a social network. Seven users work on common projects, and two users are connected if they work on a common project (i.e., a two-mode or affiliation network; see Wasserman and Faust 1999). An actor’s embeddedness increases as it connects to more actors (Swaminathan and Moorman 2009) and becomes more central to the social network (Ronchetto et al. 1989). For example, in Figure 2, user 4 works on two projects, and user 3 works on three, so user 3 is more embedded than user 4. Brokerage instead increases with actors’ ability to connect disconnected parts of the network (Burt 2005, Fleming and Waguespack 2007). User 4 brokers the relationship between two groups (left and right of user 4),

and as the sole link between them, it earns a high brokerage score by spanning gaps (Burt 2004).

2.1.1. Embeddedness. Embeddedness provides efficient access to other parts of the network (Grewal et al. 2006, Ronchetto et al. 1989), so greater founders’ embeddedness signals greater and often-required redundancy in connections. By serving as the access point for redundant connections, highly embedded founders enjoy the benefit of being the centers of action and can coordinate activities (Swaminathan and Moorman 2009). That is, high embeddedness provides the resources, information, and feedback necessary to ensure coordination in the OSS domain (Fleming et al. 2007, Kogut and Zander 1992), and it thereby reduces the time to product release (Barczak et al. 2009).

In our research context, newly founded projects compete in a crowded virtual community with strict resource constraints, with high rates of failure (von Krogh and von Hippel 2006). Projects initiated by highly embedded founders gain access to diverse member skills, enjoy the benefits of improved coordination (Uzzi 1997), and experience quicker product development. However, greater founder embeddedness means that they can devote less attention to any given project (Rosa et al. 1999). Therefore, embeddedness should decrease time to market, with diminishing returns, and we hypothesize the following.

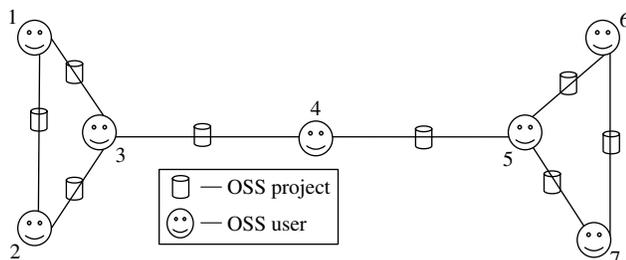
HYPOTHESIS 1 (H₁). *As the project founders’ embeddedness at the time of founding increases across projects, the time to product release decreases at a decreasing rate.*

2.1.2. Brokerage. Developer users that broker relationships by linking sparsely connected parts of a social network have access to and control over information, which gives the broker power and fosters creativity (Burt 2004, 2005). The combination of power and creativity should result in timely completion of high-quality NPD projects (Fleming et al. 2007, Nerkar and Paruchuri 2005). The existing NPD literature supports this assertion for the general NPD process (e.g., Sethi et al. 2001, Song and Parry 1997). For OSS, such improvements should result in shorter mean times to product release with diminishing returns to social capital (Zucker et al. 1995). Thus, we hypothesize the following.

HYPOTHESIS 2 (H₂). *As the project founders’ brokerage at the time of founding increases across projects, the time to product release decreases at a decreasing rate.*

2.1.3. Interaction of Embeddedness and Brokerage. Several studies predict trade-offs in the extent to which social network members benefit from various types of social capital (Burt 1997, Obstfeld 2005, Fleming et al. 2007). For example, Burt (2004) shows

Figure 2 Illustration of Brokerage and Embeddedness



Notes. (1) User 1 shares a project with users 2 and 3; user 2 shares a project with users 1 and 3; user 3 shares a project with users 1, 2, and 4; user 4 shares a projects with users 3 and 5; user 5 shares a project with users 4, 6, and 7; user 6 shares a project with users 5 and 7; and user 7 shares a project with users 5 and 6. (2) User 4 works on two projects, whereas user 3 works on three projects, so user 3 is more embedded than user 4 (degree centrality; see Ronchetto et al. 1989). (3) User 4 brokers the relationship between the two groups, one comprising users 1, 2, and 3 and the other users 5, 6, and 7. As the sole link between these otherwise disconnected groups, user 4 scores higher on brokerage than the other users.

that decreased brokerage might reflect greater connection density for a member, a larger network, or changes in the network hierarchy. Therefore, members in a network structure may have to shift their location to gain the benefits of one type of social capital relative to another. For our study context, this trade-off refers to the project founder's desire to achieve efficiency by improving embeddedness over the desire to foster creativity or increase brokerage by occupying a unique network location (Fleming et al. 2007). It parallels the exploration versus exploitation dichotomy that March (1991) proposes for organizational learning processes.

Because founders' connections facilitate information and resource dissemination, greater embeddedness should directly benefit their new projects. When brokerage increases, founders also should obtain more indirect value from being able to combine disconnected sets of resources through greater embeddedness, which facilitates information dissemination (Fleming et al. 2007). Then projects benefit from founders' ability to create an environment marked by both explorative and exploitative processes (Obstfeld 2005). We predict that embeddedness has a greater impact on time to product release when brokerage increases.

HYPOTHESIS 3 (H₃). *As the project founders' brokerage at the time of founding increases across projects, the magnitude of the reduction in time to product release as a result of embeddedness increases.*

2.2. Interplay of Developer Users and End Users

The exchange of information between developer user and end user communities is the primary basis for the creation and maintenance of user-generated content; we characterize this exchange using the degree of user engagement and the product's audience focus.

As user engagement (i.e., number of bugs reported) increases, the benefits that the project team receives from the community's feedback also increase (Grewal et al. 2006). Products that garner many bug reports benefit from elevated user activity, which reveals problems in functionality and suggests fixes. Overall, the degree of user engagement should reduce time to product release. However, when the project founder's social capital is greater, the beneficial effects of degree of user engagement should be stronger: More social capital enables founders to understand and take advantage of user engagement and feedback to speed up the NPD process (Grewal et al. 2006, Hahn et al. 2008). Therefore,

HYPOTHESIS 4 (H₄). *As the degree of user engagement increases across projects, the magnitude of the reduction in time to product release as a result of the founders' (a) embeddedness and (b) brokerage increases.*

The product's audience focus is critical in the OSS context (Subramaniam et al. 2009). The project

administrator decides whether to target developer users (technical) or end users (nontechnical), although it seems logical that development teams prioritize products with the broadest community appeal (Lakhani and von Hippel 2002, Shah 2006). Therefore, user-focused, nontechnical products should exhibit a shorter time to product release than developer-focused, technical products. The effect of the founder's social capital also should vary by the product's audience focus. User-focused projects may generate less technical feedback, resulting in faster development (Lakhani and von Hippel 2002). That is, the reduction in time to product release as a result of social capital should be greater for user-focused compared with developer-focused projects.

HYPOTHESIS 5 (H₅). *The magnitude of the reduction in time to product release across projects as a result of the founders' (a) embeddedness and (b) brokerage is greater for user-focused projects than for developer-focused products.*

3. Model Development

To model the effect of the three theorized factors on the time to product release of user-generated products, we denote the founding point as time 0 and seek a model that accounts for (1) censored observations, (2) projects that never release a product, (3) the influence of covariates (some of which vary over time) and control variables, (4) unobserved heterogeneity, and (5) an appropriate specification of the underlying timing distribution. In our base model, the random variable T represents the time to the first product release when the hazard rate—or the probability of product release at time T , given that the first release has not occurred—is $h(T) = f(T)/S(T)$, with T as the time to product release, $f(\cdot)$ as a probability density, and $S(\cdot)$ as a survival function, where $S(\cdot) = 1 - F(\cdot)$ and $f(\cdot) = F'(\cdot)$, with $F(\cdot)$ as a cumulative distribution function.

3.1. Censored Observations

Some covariates that affect the hazard vary over time, so for each project i in the sample, the interval $(0, t_i]$ is divided into k exhaustive, nonoverlapping subintervals, such that $t_0 < t_1 < \dots < t_{k-1} < t_k$, with $t_0 = 0$ and $t_k = t_i$. Then C_i indicates the release event for project i ; so $C_i = 1$ if a release occurs and 0 otherwise. The covariate vector is $X_i = (X_i^{\text{DU}}, X_i^{\text{EU}}, X_i^{\text{C}})$, where X_i^{DU} represents the project founders' social capital variables derived from the developer user network, X_i^{EU} represents variables that characterize the interplay between developer users and end users, and X_i^{C} is the OSS project and product characteristics, i.e., control variables. For simplicity, we drop the subscript i . Covariates may change from one interval to the next but stay constant within an interval. The

hazard function from time t_{j-1} to t_j can be written as $h(t|X_j)$, and because $h_t = -d \log S(t)/dt$,

$$\Pr[T < t_j | T \geq t_{j-1}] = \exp\left(-\int_{t_{j-1}}^{t_j} h(s | X_j) ds\right). \quad (1)$$

Therefore, the survival function for a duration of t_k or longer can be written as

$$\begin{aligned} S(t_k|X_k) &= \prod_{j=1}^k \Pr[T \geq t_j | T \geq t_{j-1}] \\ &= \prod_{j=1}^k \exp\left(-\int_{t_{j-1}}^{t_j} h(s | X_j) ds\right). \end{aligned} \quad (2)$$

If we take the logs of both sides of Equation (2), we derive

$$\ln(S(t_k)) = \sum_{j=1}^k \int_{t_{j-1}}^{t_j} h(s | X_j) ds. \quad (3)$$

The log-likelihood for a single observation can be written as the sum of the contributions to the hazard and survival functions; that is,

$$LL_i = C_i \cdot \ln(h(t_k | X_k)) - \sum_{j=1}^k \int_{t_{j-1}}^{t_j} h(s | X_j) ds. \quad (4)$$

Finally, projects that do not get released during the observation window $(0, T_c]$, such that $C_i = 0$ at the end of the observation window, contribute only to the survival function. Projects released during one of the k intervals during the observation window $(0, T_c]$, such that $C_i = 1$, contribute to both survival and hazard functions. If we substitute Equation (3) into Equation (4), we can simplify the log-likelihood as

$$LL_i = C_i \cdot \ln(h(t_k | X_k)) + \ln(S(t_k)). \quad (5)$$

3.2. Accounting for Projects That Never Release

A standard assumption in basic hazard models is that all observations eventually experience the event, which implies an eventual occurrence probability of 1. However, this probability rarely equals 1 for all observations (Schmidt and Witte 1989), because some open source projects may never release a product file. To account for two subpopulations (events that occur and those that do not), we modify the basic hazard specification with a split-hazard formulation (e.g., Dekimpe et al. 1998, Sinha and Chandrashekar 1992); the probability of eventual product release δ_i is a function of covariate vectors, including founders' social capital variables X_i^{DU} , characteristics of the interplay between communities X_i^{EU} , and OSS project and product characteristics X_i^C , as $\delta_i = \delta(X_i^{DU}, X_i^{EU}, X_i^C)$. The log-likelihood for a single observation can be specified as

$$LL_i = C_i \cdot \ln(h(t_k | X_k)) + \ln(1 - \delta_i + \delta_i \cdot S(t_k)). \quad (6)$$

3.3. Model Covariates and Controls

We incorporate the effects of project founders' social capital (X_i^{DU}), the characteristics of the interplay between developer users and end users, i.e., the moderators (X_i^{EU}), and OSS project and product characteristics (X_i^C) on time to product release. We also incorporate the interaction between the moderators and founder's social capital. Let λ_i be the positive location parameter, and let p be the scale parameter of the duration distribution. We incorporate these effects by modifying the location parameter of the hazard function λ_i as a function of X_i^{DU} , X_i^{EU} , and X_i^C as follows:

$$\lambda_i = \exp(\alpha_0 + \beta' X_i^{DU} + \gamma' X_i^{EU} + \kappa' X_i^{DU} X_i^{EU} + \nu' X_i^C), \quad (7)$$

where α_0 is the constant term, β' captures the effect of founders' social capital variables, γ' is the main effect of moderators (i.e., degree of user engagement and product's audience focus), κ' refers to the interaction between social capital variables X_i^{DU} and the moderators X_i^{EU} , and ν' is the effect of control variables X_i^C . The degree of user engagement is a time-varying covariate in the model.

In addition, to accommodate the effect of unobserved individual project characteristics, we incorporate the effect of unobserved heterogeneity, using a random effects approach, with a random parameter (Sastry 1997). Thus, a standard normal random variable $\varepsilon \sim N(0, \sigma^2)$ affects the location parameter as follows:

$$\begin{aligned} \lambda_i(\varepsilon_i) &= \exp(\alpha_0 + \beta' X_i^{DU} + \gamma' X_i^{EU} \\ &\quad + \kappa' X_i^{DU} X_i^{EU} + \nu' X_i^C + \sigma \varepsilon_i). \end{aligned} \quad (8)$$

Then the modified log-likelihood conditional on ε_i can be specified as

$$\begin{aligned} LL | \varepsilon_1, \dots, \varepsilon_N &= \sum_{i=1}^N [C_i \cdot \ln(h(t_i | \lambda_i(\varepsilon_i), p)) \\ &\quad + \ln(1 - \delta_i + \delta_i \cdot S(t_k))]. \end{aligned} \quad (9)$$

The timing process model in Equation (9) incorporates heterogeneity as a function of project characteristics on both time to product release and the probability of eventual product release (Dekimpe et al. 1998). To obtain the unconditional log-likelihood, we integrate ε_i from Equation (9), such that

$$\begin{aligned} LL &= \sum_{i=1}^N \int_{-\infty}^{+\infty} [C_i \cdot \ln(\delta_i \cdot f(t_i | \lambda_i(\varepsilon_i), p)) \\ &\quad + \ln(1 - \delta_i + \delta_i \cdot S(t_k))] f(\varepsilon_i) d\varepsilon_i. \end{aligned} \quad (10)$$

Because the log-likelihood in Equation (10) does not have a closed form, the integral can be computed by simulating draws of ε_i , then averaging the

log-likelihood across draws (Train 2003). The simulated log-likelihood approach has been employed as an effective estimation technique in other studies (e.g., Erdem and Keane 1996, Park and Gupta 2009, Villas-Boas and Winer 1999) with estimation problems similar to ours. The simulated log-likelihood is

$$LL_s = \sum_{i=1}^N \frac{1}{R} \sum_{r=1}^R \{C_i \cdot \ln(\delta_i \cdot f(t_i | \lambda_i(\varepsilon_{ir}), p)) + \ln(1 - \delta_i + \delta_i \cdot S(t_k))\}, \quad (11)$$

where ε_{ir} is a simulated random sample of R Halton draws.

3.4. Timing Distribution Specification

The timing process model in Equations (1)–(11) can accommodate a range of timing distributions, such as exponential with a constant hazard, Weibull with a monotonic hazard, and log-normal with a nonmonotonic hazard. We have no a priori theoretical reason to support one of these distributions over others and use model fit to select among them.

4. Research Context and Data

4.1. SourceForge.net: Data Context

SourceForge.net is a collaborative product development platform that enables users to coordinate their open source product development efforts (Hahn et al. 2008). As of September 2011, more than 2.7 million users worked on approximately 311,121 open source projects, including database tools, application software, games, text and programming editors, and utility tools. With the platform's collaborative capabilities, a project founder can advertise and recruit volunteer users and organize product development through the Internet. The source code of projects is publicly available; registered users can download and use or modify it to develop new versions as long as they adhere to the product's open source user license. Prior studies also employ data from SourceForge (e.g., Grewal et al. 2006, Hahn et al. 2008), and von Krogh and von Hippel (2006) cite it as an attractive venue for research into user communities and hybrid innovation models.

We gathered data on the network structure of OSS projects at their founding and the projects' characteristics from a sample of open source projects registered on SourceForge. Our access to these data proceeded through the SourceForge data warehouse, which records and organizes all activities on the site (Madey 2005). The data warehouse is a separate entity from the website and provides a repository of activities taking place on SourceForge, maintained by researchers at the University of Notre Dame.

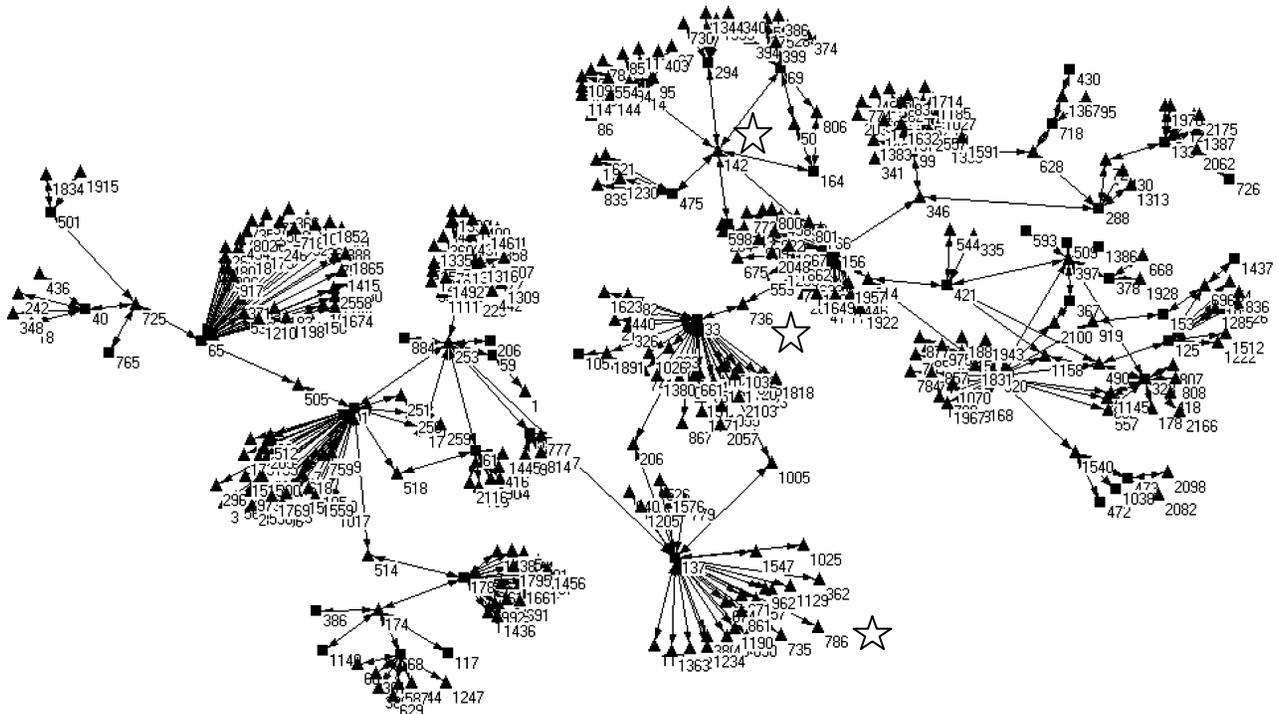
To avoid inconsistencies or confounds from different start times and obtain the longest observation possible, we only selected new projects initiated during the first 10 days of January 2003, the earliest available data. We identified 817 new projects, which we tracked for 42 months. The first file release by the project, recorded in the data warehouse, is our primary interest. We tabulated such events for our sample and calculated the time to first release (registration to release) in hours. We found 468 projects that resulted in product releases; 349 projects did not report a product release during the 42-month observation window.

4.2. The Network of User-Generated Products and Founders

Because data on the network structure of the projects are not directly available on SourceForge, we constructed the network structure at founding from user-project membership information. The data warehouse stores monthly snapshots of all activity on the SourceForge website, so we accessed database tables for January 2003, which represent the cumulative history from 1999 to 2003. Subsequent monthly data dumps are available beginning November 2004 and are updated every month. These data about user-project memberships provided details about the founding users for 817 focal projects in the user-project membership table for January 2003. We then transformed the relational data (i.e., who founded which project) into network structure data, represented by an affiliation network matrix (Wasserman and Faust 1999) with the network analysis software UCINET 6. An affiliation network matrix depicts actors (i.e., users) as rows and events (i.e., projects) as columns; each cell takes a value of 1 if the user works on a project or 0 if the user does not (see Online Appendix A in the electronic companion, available as part of the online version that can be found at <http://mktsci.journal.informs.org/>).

Because our objective is to study newly founded projects initiated in a connected network, we applied a snowballing procedure (Wasserman and Faust 1999) in which we listed all projects whose development efforts formally initiated in the first 10 days of January 2003, and we then listed all 966 developer users working on these initiatives. We next listed other (existing) projects on which these 966 developer users had worked, excluding the first set of 817 projects, and identified 624 additional projects. Finally, we listed all developer users working on these 624 projects and thereby identified 1,723 additional individuals. This procedure produced a network of 1,441 projects and 2,689 developer users. We again transformed these relational data into an affiliation network, in which a 1 in a cell indicates that the user identified by the column works as a contributor on the project identified by the row and a 0 indicates the absence of such

Figure 3 Representation of the Network Sample of Open Source Projects and Users



Notes. This network graph was created using Pajek software (Batagelj and Mrvar 1998). The squares represent newly founded open source projects ($N = 817$), and the triangles are users. Consider users 142, 736, and 786 (marked by stars). User 736 is connected to two projects (33, 156), and the sole connection for otherwise unconnected network parts is an indicator of its large brokerage measure. User 142 is highly embedded as part of multiple projects (475, 95, 294, 69, 164, 598). User 786 is connected to just one project (178) on the periphery of the network and has low brokerage and embeddedness.

a relationship. This matrix captures all relationships defined by the founders of the original 817 new initiatives. We present a visual plot of the largest continuously connected component of the network sample in Figure 3.

5. Measures

5.1. Time to Product Release

We use the time until the first release of the product as our dependent variable. Of the 817 projects, 468 (57.3%) experienced a product release and thus provide an observed time to product release. We treat the remaining 349 projects (42.7%) as right-censored observations. With hours as the unit of time, we employ a continuous time modeling approach; the dependent variable is a time measure and can only take positive values, so we log transform it (Kalbfleisch and Prentice 1980, Kiefer 1988, Sinha and Chandrashekar 1992).

5.2. Project Founders' Social Capital

Scholars use various embeddedness operationalizations, depending on the context, but the most widely employed measures include degree centrality, betweenness centrality, and closeness centrality

(Ronchetto et al. 1989); we employ all three. *Degree centrality* is the number of existing projects connected to any given project founder in the network; in our study context, it captures the project founders' diverse contacts in the network. *Betweenness centrality* measures how well the network member can mediate flows of information; it is a count of all the shortest possible paths that include a founder and represents the ease with which the project founder can facilitate a transfer of resources over the network. To assess the efficiency of access to other locations by the shortest paths, we use *closeness centrality*. This measure is the sum of the shortest paths between the project founder and all other projects in the network. Greater closeness means the founder relies more on other members in the network (Ronchetto et al. 1989). Because closeness centrality is a summed measure, after being normalized, a higher value means the project founder is less embedded; we use the inverse of this summed measure.

To measure *brokerage*, we follow prior research (Ahuja 2000, Fleming and Waguespack 2007) and apply a version of what Burt (1992) calls "constraint," or the degree to which a network member depends on directly connected neighbors to connect to others in the network. Therefore, brokerage measures

the degree of opportunity available to the founder to form new connections in the network. Higher constraint implies lower brokerage, so we inverse code this measure by subtracting it from 1. When a project has multiple founders, we use an average to develop a composite measure. We provide the intuition underlying the measures in Figure 2 and the technical details regarding their calculation in Appendix A of the electronic companion.

5.3. Interplay of Developer Users and End Users

The number of bugs reported provides our measure of the *degree of user engagement* (Grewal et al. 2006). This measure is highly correlated with the number of bugs resolved and thus captures the degree of product quality to a large extent, because resolved bugs signal the degree to which problems in the product have been addressed. A more engaged user community can benefit the development process and lead to better products (Crowston et al. 2003, Lee et al. 2009). The number of reported bugs varies over time, so we obtained the data at six distinct times: November 2004, February 2005, June 2005, October 2005, February 2006, and June 2006. This variable is the time-varying covariate in the model.

Following prior research (Subramaniam et al. 2009), we also characterize projects according to the *product's audience focus* as developer-focused or user-focused. We constructed a dichotomous measure from information in the project details table in the data warehouse, such that 1 indicates a user-focused project and 0 a developer-focused project.

5.4. Control Variables

Project founders develop coordination and product development skills through interactions in the open source community (O'Mahony and Ferraro 2007), and greater experience might enable them to foresee and respond to problems better (Chandrashekar

et al. 1999), which would benefit product development. Therefore, we measure *founders' experience* as the cumulative time since they registered in the open source community. When a project has multiple founders (34% of sample projects), we sum their experience. We also obtained data about whether the project development process relies on the community forum to communicate with users (Madey 2005). This control variable, *forum usage*, indicates the extent of use of the community feature provided by the website. Projects that use the forum may benefit more than projects that do not, so we control for this effect (1 if the project uses a forum and 0 if not). In the OSS context, different licenses grant different degrees of control over the source code (Stewart et al. 2006, Subramaniam et al. 2009). The most common include the general public license (GPL) and the less restrictive limited general public license (LGPL). Because GPL is the most widely used, we code the *type of OSS license* as a dichotomous variable, where 1 indicates use of GPL and 0 otherwise. Finally, we include variables that measure the size of the released project files (*file size at release*) and number of project members (*team size*). Some projects lead to product releases immediately after registration because the development team has been working on the source code before registering. To control for *prior code availability*, we include a dichotomous variable that measures whether prior code was available at the time of registration (e.g., Subramaniam et al. 2009), which equals 1 if a prior code was available and 0 otherwise.

5.5. Descriptive Statistics

Table 1 provides descriptive statistics. The mean for product's audience focus is 0.52, indicating that 52% of projects in our sample were user-focused. The mean for type of license is 0.33, indicating that 33% of projects used the GPL. A mean of 0.15 for prior code availability indicates that 15% of the projects had

Table 1 Descriptive Statistics and Bivariate Correlations (*n* = 817)

Variable	Mean	SD	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
(1) <i>Brokerage</i>	0.30	0.33	1												
(2) <i>Degree centrality</i>	2.03	3.55	-0.34	1											
(3) <i>Betweenness centrality</i>	48.35	266.02	-0.14	0.18	1										
(4) <i>Closeness centrality</i>	0.03	1	-0.62	0.43	0.19	1									
(5) <i>Product's audience focus</i>	0.52	0.50	0.09	-0.03	0.01	-0.07	1								
(6) <i>Type of OSS license</i>	0.33	0.47	-0.09	0.05	0.07	0.07	-0.08	1							
(7) <i>Prior code availability</i>	0.15	0.36	0.01	-0.02	-0.05	0.03	-0.01	-0.05	1						
(8) <i>Forum usage</i>	0.89	0.31	-0.09	0.04	0.04	0.08	-0.01	0.03	-0.10	1					
(9) <i>Bugs reported</i>	3.55	42.24	0.02	-0.01	-0.01	-0.02	0.03	-0.02	-0.01	-0.13	1				
(10) <i>Project team size</i>	1.82	2.27	-0.06	0.03	-0.03	-0.01	-0.09	0.17	-0.03	-0.02	-0.01	1			
(11) <i>Founder's experience</i>	2683.63	1303.23	-0.23	0.14	0.08	0.34	0.03	0.05	0.07	-0.00	-0.04	0.01	1		
(12) <i>File size at release</i>	1207.13	18,799,094	-0.09	0.06	0.06	0.07	0.02	0.08	0.02	0.02	-0.03	0.07	-0.04	1	
(13) <i>Time to product release</i> ^a	487.34	7.956	-0.12	-0.21	-0.04	-0.09	-0.01	0.04	-0.05	0.02	0.02	-0.44	-0.04	-0.03	1

^aTo calculate the mean for this variable, we only used those projects that released a product during the observation window.

INFORMS holds copyright to this article and distributed this copy as a courtesy to the author(s).
 Additional information, including rights and permission policies, is available at <http://journals.informs.org/>.

a prior code at the time of registration, and a mean of 0.89 for forum usage indicates that 89% of projects used the community forum. The average project had 3.55 bug reports during the observation window, with a team size of 1.82 at the end of the first month (founders and non-founders) and 112 days of project founders' experience. Of the projects that released a product file, the average file size at release was 0.44 MB, released after 487 hours, or a time to product release of roughly 20 days.

6. Results

6.1. Model Selection

Following Sinha and Chandrashekar (1992), we use a two-step procedure to select the appropriate model functional form (see Appendix B of the electronic companion). First, we seek an appropriate functional form for the timing distribution; we investigated the hazard function using Kaplan–Meier estimator, which suggests the hazard is nonmonotonic. Thus, we ruled out the exponential and the Weibull models, as these have constant and monotonic hazard functions, respectively. We then tested the log-logistic, log-normal, and the gamma specifications, all of which accommodate nonmonotonic hazards. The fit criteria indicated that the model with a log-logistic distribution (model M2: $BIC_{M2} = 3,958.92$, $CAIC_{M2} = 3,991.92$)² fit the duration data better than the model with the log-normal distribution (M1: $BIC_{M1} = 4,375.70$, $CAIC_{M1} = 4,408.70$) or the gamma distribution (M3: $BIC_{M3} = 3,983.37$, $CAIC_{M3} = 4,017.37$).

Second, in the log-logistic model, a comparison of a probit (M4) versus a logit (M5) specification for the probability of eventual product release in the split-hazard specification (Schmidt and Witte 1989, Sinha and Chandrashekar 1992) shows that the logit specification fits the data better (M5: $BIC_{M5} = 3,187.46$, $CAIC_{M5} = 3,942.46$; M4: $BIC_{M4} = 3,797.78$, $CAIC_{M4} = 3,861.78$). We also find a statistically significant improvement in this logistic distribution specification for the hazard with logit link function specifications for the probability when we account for unobserved heterogeneity in the model (M6). The final model (M6) outperforms the best model from the second stage (M5) on various fit criteria ($BIC_{M6} = 3,796.16$, $CAIC_{M6} = 3,861.16$). We estimated M6 using simulated maximum likelihood estimation with a sequence of 30 Halton draws instead of the standard normal random draws³ (Bhat 2003, Train 2003). In Table 2, we present the results from this final model.

² BIC is the Bayesian information criterion; CAIC is the consistent Akaike information criterion.

³ Bhat (2003) finds, in a simulation study, that Halton draws increase processing speed without producing significant effects on

6.2. Probability of Eventual Product Release

The results from the logit portion of the split-hazard formulation reveal which projects eventually lead to a product release (see Table 2). As degree centrality increases, the probability of eventual product release increases at an increasing rate (positive quadratic term; $b = 0.04$, $p < 0.01$); as closeness centrality increases, this probability increases (positive linear term: $b = 0.61$, $p < 0.10$). As brokerage increases, the probability of eventual release increases at a decreasing rate (positive linear effect: $b = 1.14$, $p < 0.01$; negative quadratic effect: $b = -0.54$, $p < 0.01$). The embeddedness measures pertaining to betweenness ($b = 0.17$, $p < 0.01$) and closeness ($b = 0.83$, $p < 0.01$) centrality strengthen the positive effect of brokerage on the probability of eventual product release.

Developer-focused projects are more likely to experience product release than are user-focused projects ($b = -0.69$, $p < 0.01$), though projects that use a GPL are less likely to do so ($b = -0.33$, $p < 0.05$). The probability of eventual product release increases with greater founders' experience ($b = 0.21$, $p < 0.01$), and using a community forum increases the probability of eventual product release ($b = 0.60$, $p < 0.05$). The results for the interplay of developer user and end user communities reveal that increasing brokerage increases the probability of eventual release at a faster rate for end user than for developer user products ($b = 0.21$, $p < 0.10$).

In summary, the results indicate that the social capital of the project founders significantly impacts the probability of eventual product release. Projects that are developer-focused, use community forums, and were started by experienced founders are also more likely to release products eventually than projects that are user-focused, do not use forums, or were started by less experienced founders.

6.3. Time to Product Release

6.3.1. Project Founders' Embeddedness and Brokerage. In support of H_1 , time to product release decreases at a decreasing rate across projects as embeddedness increases, where embeddedness is measured by degree centrality (negative linear effect: $b = -0.10$, $p < 0.10$; positive quadratic effect: $b = 0.01$, $p < 0.01$) and closeness centrality (negative linear effect: $b = -0.56$, $p < 0.05$; positive quadratic effect: $b = 0.39$, $p < 0.01$). In contrast, time to product release increases at a decreasing rate as betweenness centrality increases (positive linear effect: $b = 0.01$, $p < 0.01$; negative quadratic effect: $b = -0.02$, $p < 0.01$). The

the simulation results; the results are robust to a number of Halton draws that is slightly higher than the square root of the number of observations. We use 30 draws for our sample size of 817.

Table 2 Effects of Social Network Variables and Product Characteristics ($n = 817$)

	Time to product release	Probability of eventual product release
Constant	6.89*** (1.92)	4.36 (5.74)
Founder's social capital		
Embeddedness (H_1)		
Degree centrality (DC)	-0.10* (0.07)	-0.04 (0.13)
DC^2	0.01*** (0.003)	0.04*** (0.01)
Betweenness centrality (BC)	0.01*** (0.02)	0.01 (0.01)
BC^2	-0.02*** (0.01)	-0.12 (0.133)
Closeness centrality (CC)	-0.56** (0.33)	0.61* (0.47)
CC^2	0.39*** (0.17)	-0.08 (0.11)
Brokerage (H_2)		
Brokerage (B)	-0.92*** (0.18)	1.14*** (0.36)
B^2	0.50*** (0.14)	-0.54*** (0.11)
Interplay of developer users and end users		
Degree of user engagement		
Bugs reported (BR)	-0.05*** (0.02)	0.05 (0.04)
Product's audience focus (PA)	0.22 (0.22)	-0.69** (0.18)
Interaction between embeddedness and brokerage (H_3)		
$DC * B$	-0.42*** (0.18)	0.18 (0.21)
$BC * B$	0.02 (0.03)	0.17*** (0.01)
$CC * B$	-0.68*** (0.19)	0.83*** (0.23)
Interactions of founder's social capital with interplay of developer users and end users (H_4)		
$DC * BR$	-0.09*** (0.04) ^a	-0.02 (0.03)
$BC * BR$	-0.05** (0.03)	-0.00 (0.00)
$CC * BR$	-0.01 (0.03)	0.06 (0.10)
$B * BR$	0.03 (0.02)	0.15 (0.34)
(H_5)		
$DC * PA$	-0.70*** (0.20) ^a	0.01 (0.01)
$BC * PA$	0.35** (0.21)	-0.21 (0.24)
$CC * PA$	-0.25* (0.17)	0.06 (0.23)
$B * PA$	-0.01 (0.06)	0.21* (0.13)
Control variables		
Experience of founders	-0.15*** (0.02)	0.21*** (0.03)
Forum usage	-0.80*** (0.34)	0.60** (0.35)
Type of OSS license	-0.01 (0.20)	-0.33** (0.18)
File size at release	0.00 (0.001)	NA
Project team size	0.00 (0.001)	0.03 (0.04)
Prior code availability	-0.69*** (0.27)	0.46 (0.31)
Unobserved heterogeneity parameter	-0.17*** (0.08)	NA
Scale parameter of duration distribution	1.04*** (0.05)	NA

^aThe coefficient and standard error were multiplied by 10^2 for ease of presentation.

* $p < 0.1$; ** $p < 0.05$; *** $p < 0.01$.

findings related to betweenness centrality are consistent with other studies that show varying results concerning the strength and direction of relationships involving betweenness centrality (Ronchetto et al. 1989, Fleming and Waguespack 2007). Because betweenness centrality measures a founder's location

on the shortest pathways in the network, it is possible that information overflow along the shortest pathways may lead to information overload (Rosa et al. 1999). In that case, founders occupying such locations can slow down both the NPD process and the time to product release.

Consistent with H_2 and extant research on diminishing returns to (social) capital (Elfenbein and Zenger 2009, McFadyen and Cannella 2004), as brokerage increases, the time to product release decreases at a decreasing rate (negative linear effect: $b = -0.92$, $p < 0.01$; positive quadratic effect: $b = 0.50$, $p < 0.01$). Thus, brokerage offers benefits, but the rate of return decreases as the level of brokerage increases.

6.3.2. Interaction of Embeddedness and Brokerage. In support of H_3 , regarding the synergy between embeddedness and brokerage in OSS product development, we find that the magnitude of the decrease in the time to product release with an increase in embeddedness increases as brokerage increases. However, this effect is only significant for embeddedness measured by degree centrality ($b = -0.42$, $p < 0.01$) and closeness centrality ($b = -0.68$, $p < 0.01$). The significance of the interaction between embeddedness and brokerage implies that projects whose founders are endowed with both high embeddedness and high brokerage see first releases quicker than projects founded by people who score high only on one of the social capital constructs.

Ai and Norton (2003) indicate that in nonlinear models, the parameter value of an interaction effect does not appropriately reflect the marginal interaction effect. Following Greene (2010),⁴ in addition to the statistical significance of the interaction effects presented here, we also present a detailed analysis of the interactions in our model in the appendix. These additional analyses confirm the conclusions drawn from the coefficients of the interaction terms.

6.3.3. Interplay of Developer Users and End Users. As the degree of user engagement for a project increases,⁵ time to product release decreases ($b = -0.05$, $p < 0.01$). Consistent with H_{4a} , we find a decrease in the time to product release as the degree of user engagement increases when embeddedness increases (degree: $b = -0.09$, $p < 0.01$; betweenness: $b = -0.05$, $p < 0.05$). Thus, our findings support the conjecture that the developer user community leverages the activity of the end user community in the OSS context.

⁴ Greene (2010, p. 295) notes that “partial effects are neither coefficients nor elements. . . . They are implications of the specified and estimated model. . . . [t]he most informative point in the analysis at which to do hypothesis testing is at the model building step, not at the analysis step.” As partial effects are an artifact of the functional form, we test hypotheses using the interaction effects estimated in Table 2. However, to understand the nature of the interactions, we examine both the partial interaction effects and the effect on the dependent variable visually in the appendix (Greene 2010).

⁵ We included the residual error term from the regression of bugs reported on bugs resolved as an explanatory variable in an alternative analysis. The variable instruments the project team's innate ability to resolve bugs above and beyond bugs reported. The effect of the residual error term was -0.02 ($p < 0.05$).

Consistent with H_{5a} , the shortened time to product release as embeddedness increases across projects becomes even shorter for user-focused projects than for developer-focused projects (degree: $b = -0.007$, $p < 0.01$; closeness: $b = -0.25$, $p < 0.10$). In contrast, when betweenness centrality increases, the shortening of the time to product release is greater for developer-focused projects than for user-focused projects ($b = 0.35$, $p < 0.05$).

In summary, founders' embeddedness seems more relevant than brokerage in the interaction with degree of user engagement and product's audience focus.

6.3.4. OSS Project and Product Characteristics. As the project founders' experience increases, the time to product release decreases ($b = -0.15$, $p < 0.01$), and projects that use community forums experience a shorter time to product release than projects that do not ($b = -0.80$, $p < 0.01$). Projects with a prior code at the time of registration on SourceForge release products at a faster rate than projects without this code ($b = -0.69$, $p < 0.01$). Finally, the effects of the type of open source license, size of first release file, and size of project team are insignificant.

6.4. Robustness Checks and Model Validation

6.4.1. Sampling Frame. To rule out the possibility that our results are specific to our sampling frame, we created three random subsets from the 817 projects that constitute our sample. We then created three different samples by combining two random samples at a time: subsample 1 ($n = 582$), subsample 2 ($n = 510$), and subsample 3 ($n = 542$). We estimated three split-hazard models with unobserved heterogeneity, one for each subsample. The results from all three models are consistent with those from the entire sample.

6.4.2. Predictive Validity. To assess the predictive validity of the duration model, we divided the sample into two random subsets, A for estimation ($n = 410$, censored = 177) and B for predictive validation ($n = 407$, censored = 172). We estimated two models using the sample A data: one with only project feature variables and another that includes both the social capital variables and the interplay of the developer user and end user communities. We account for censoring by following Grewal et al. (2004) and removing censored observations from the holdout sample before calculating the predictive validity scores. With the sample B data, we compared the two models in terms of their root mean square errors (RMSE), mean absolute errors (MAE), and Thiel's U statistic (TUS) (Greene 2003). The model that includes the network measures provides superior predictions than the one without them (RMSE improves by 12.4%, MAE by 17.8%, and TUS by 11.3%).

We validate the predictive ability of the eventual product release (i.e., logit part of the split hazard model) for the overall sample (Sinha and Chandrashekar 1992). With a correct prediction rate of 0.82,⁶ our model specification outperforms the naïve criteria of $C_{\text{pro}} (= 0.51)$ and $C_{\text{max}} (= 0.57)$ (Morrison 1969).⁷

6.4.3. Alternative Dependent Measures. We conducted post hoc analyses of downloads and page views for projects in our sample. The time to first release, as expected, is negatively correlated with the cumulative number of downloads (-0.39) and page views (-0.19) before the first release date. The more the working version is downloaded or the more page views it garners, the shorter the time to release, which validates our premise that success measures such as downloads correlate with our dependent variable and in the right direction. We also ran two regression models, with downloads or page views as the dependent variable. We find that the explanatory variables that increase downloads also decrease time to product release. For example, project founder's degree centrality has a negative effect on time to product release (-0.10 , $p < 0.10$) but increases downloads (0.01 , $p < 0.01$) and page views (0.03 , $p < 0.01$). Similarly, brokerage decreases time to product release (-0.92 , $p < 0.01$), whereas it increases both downloads (0.66 , $p < 0.05$) and page views (0.04 , $p < 0.05$).

7. Discussion

As the economic significance of user-generated content continues to grow, it becomes critical to understand how such content is created, managed, and used in various contexts. We study the factors that drive the success of new OSS projects, an important type of user-generated content. We conceptualize the interplay of two communities, developer users and end users, as the necessary basis for the creation and maintenance of user-generated content. Therefore, the time to product release of new OSS projects is influenced by the project founders' social capital, aspects of the interplay of developer user and end user communities, and the OSS project and product characteristics.

7.1. Theoretical Contributions

In Table 3, we summarize our study's findings with respect to extant research in innovation, social networks, and OSS product development. Several of our

hypotheses are new, and our findings add to sparse research on specific factors that influence NPD success in an OSS context. Grewal et al. (2006) consider the role of networks in the technical and commercial success of OSS products, and Hahn et al. (2008) examine how prior ties affect the formation of new OSS project teams. Other studies investigate the role of OSS project features on success (Stewart et al. 2006, Subramaniam et al. 2009). We extend this body of literature by studying the effect of all three factors simultaneously on time to product release.

Emerging literature in OSS also recognizes the importance of social networks (Hahn et al. 2008); we extend this research in two ways. First, rather than studying product outcomes such as commercial success (Grewal et al. 2006), we consider the NPD process with time to first product release as an NPD process metric. Second, together with embeddedness (Grewal et al. 2006), we consider brokerage, which captures a founder's access to diverse information sets and thus creativity (Burt 2004, Obstfeld 2005). Our results regarding the nonlinear main effect of brokerage and the interaction between embeddedness and brokerage (see Figure A.1) support the inclusion of both measures in a conceptualization of social capital in the OSS context.

Our work explicitly recognizes the role of two subcommunities in the network and their interplay, which is important for two reasons. First, it is central to the OSS NPD process and common in other user-generated contexts, such as pictures on Flickr and videos on YouTube. Second, it represents a contrast from the firm-centric NPD process; that is, although firms may seek user feedback during the NPD process, the level and intensity of such feedback is a managerial decision, not driven by users themselves.

7.2. Managerial Implications

Our research has particular implications for managers of firms that participate in open source projects. Project founders' social capital plays a critical role, beyond the interplay of the developer user and end user communities or the impact of OSS project and product characteristics. If a firm engaged in the OSS community were to choose between, say, users 714 and 426 (see Figure 3; actual members of SourceForge) to collaborate on a project, it should consider their embeddedness scores: user 714 is likely to bring a project to market almost a full year (357 days) before user 426 would. Similar metrics might consider user and project data, as outlined in our model. For example, projects achieve a 51% decrease in their time to product release when the founder has high embeddedness (degree centrality in particular) and brokerage, but the decrease is only 12% when the founder

⁶ We revisited the data in 2011 to verify how many of those projects eventually released a product. Of 349 censored projects, 189 remained as a work in progress, with code still available; 121 released a product; and 39 were abandoned.

⁷ C_{pro} is the total proportion of correctly classified projects that do and do not release a product, and C_{max} is the minimum expected proportion of projects that release a product.

Table 3 Summary of Results and Contributions

Hypotheses	Study findings	Contribution	Prior literature in OSS	Prior literature in social networks or innovation
H ₁ : Embeddedness decreases time to product release at a decreasing rate.	Supported (5 of 6 effects)	Established hypothesis <ul style="list-style-type: none"> • Broadened role of embeddedness to encompass its effect on a critical OSS process success measure. • Presence of a nonlinear effect. 	<ul style="list-style-type: none"> • Grewal et al. (2006): Positive effect on technical and commercial success in OSS networks. 	<ul style="list-style-type: none"> • Ronchetto et al. (1989): Positive effect on organizational buying influence in a B2B purchasing context. • Ahuja (2000): Positive effect in the context of firm innovation output. • Swaminathan and Moorman (2009): Positive effect on firm abnormal returns in marketing alliance announcements.
H ₂ : Brokerage decreases time to product release at a decreasing rate.	Supported (2 effects)	New hypothesis <ul style="list-style-type: none"> • Brokerage has a main effect on OSS product development speed. • Presence of a nonlinear effect. 	None	<ul style="list-style-type: none"> • Nerkar and Paruchuri (2005): Positive effect of brokerage in intrafirm R&D networks. • Ahuja (2000): Negative effect in the context of intrafirm networks
H ₃ : Brokerage strengthens the negative effect of embeddedness on time to product release.	Supported (2 out of 3 effects)	New hypothesis Trade-offs between types of social capital represented via interaction effects.	None	<ul style="list-style-type: none"> • Nerkar and Paruchuri (2005): Positive interaction effect between brokerage and centrality in intrafirm R&D networks.
H ₄ : Interaction between (a) embeddedness and (b) brokerage and degree of user engagement.	(a) Supported (2 out of 3 effects) (b) Not supported	New hypothesis Boundaries of social capital in the presence of interplay between developer user and end user communities.	<ul style="list-style-type: none"> • Grewal et al. (2006): Degree of user engagement as a control variable. • Subramaniam et al. (2009): Community interest as a dependent variable. 	—Not applicable, because the moderator variables are specific to the OSS context.
H ₅ : Interaction between (a) embeddedness and (b) brokerage and product's audience focus (user-focused vs. developer-focused).	(a) Supported (2 out of 3 effects) (b) Not supported	New hypothesis Boundaries of social capital in the presence of interplay between developer user and end user communities.	<ul style="list-style-type: none"> • Subramaniam et al. (2009): License and audience focus as determinants and controls of OSS success measured by user interest, developer interest, and project activity. • Stewart et al. (2006): License as a determinant of user interest and development activity. 	—Not applicable, because the moderator variables are specific to the OSS context.

Note. B2B, business to business.

has a high degree centrality alone (see Figure A.1, panel A.2). Recognizing the existence of two subcommunities helps managers find new ways to speed products to market too. They might offer more product versions targeted toward end users to encourage end user participation (e.g., beta testers) and increase bug reports. All else being equal, an increase of one standard deviation in bugs reported (42 bugs on average), or user engagement, reduces time to product release for the average project by 11%.

Finally, the effects of OSS project and product characteristics have managerial relevance. Experience pays off, and our findings help quantify that payoff; a one-standard-deviation increase (54 days on average) in the cumulative experience of project founders

reduces the time for the average project by 15%. Our results also can help managers justify the use of community collaboration tools: projects that use forums experience an 80% shorter time to product release than do projects that do not use forums.

7.3. Limitations and Research Opportunities

Our research centers on one user-generated content domain, whereas other contexts, including blogs and wikis, may not involve the same success metrics. By featuring only SourceForge, with its unique institutional idiosyncrasies, this study's propositions also require testing in other open source project domains. Our research is also limited by data availability and contains only structural network data;

we observe the connections but not what they comprise (e.g., communication patterns, formal or informal hierarchies). Information about the type and quality of communication might enrich understanding of the process. Although time to product release is a critical NPD process metric in the OSS context, other metrics such as interrelease time may also be relevant.

It would be interesting to investigate whether and how our two subcommunity characterization applies to other user-generated contexts and how our findings about the role of social capital in innovation apply more broadly to the role of social networks (Stephen and Toubia 2010). For example, bloggers typically link to other popular bloggers, and active Facebook users tend to have large social networks. Investigating the mechanics of network formation and growth in these contexts should enable managers of online social networking sites to develop new ways to manage and monetize traffic on their sites.

7.4. Conclusion

In the modern networked world, innovative models of communication and collaboration emerge rapidly, creating the need to investigate new phenomena (Reibstein et al. 2009). The emergence of user-generated content, for example, is changing how consumers share information, form communities, and connect, which has serious implications for marketing strategy. The OSS context represents one such important domain; as we show, the social network of developer users and the interplay between the developer user and end user communities have critical impacts on NPD process success. Social network analysis provides effective concepts and tools to address these new opportunities, and an emerging networked environment presents many important challenges for research. We hope our work has added insight and stimulates further research in this challenging and important domain.

Electronic Companion

An electronic companion to this paper is available as part of the online version that can be found at <http://mktsci.journal.informs.org/>.

Acknowledgments

This paper is based on the dissertation of the first author.

Appendix. Analysis of Interaction Effects

Ai and Norton (2003) indicate that in nonlinear models, the parameter value of an interaction effect does not appropriately reflect the marginal interaction effect. Following Greene (2010), in addition to the statistical significance of the interaction effects presented earlier, we also present a detailed analysis of the interactions in our model in this

appendix. To examine the interaction effects in detail, we first note that an interaction effect between two independent variables, say, v_1 and v_2 , is the change in the partial effect of v_1 on the dependent variable, say, z_1 (i.e., the partial derivative of the z_1 with respect to v_1) for a unit change in v_2 . Following Greene (2010), we investigate these effects through two types of plots—in the first we depict the partial effect, and in the second we investigate the impact on the dependent variable.

Interaction of Embeddedness and Brokerage

In panel A.1 in Figure A.1, we show how the partial effect (plotted on y axis), varies with degree centrality (x axis, 1–9 represents the sample range) across two levels of brokerage (25th versus 75th percentile), keeping all the other variables constant at their means. The changing distance between the two curves in panel A.1 shows that the partial effect differs across two levels of brokerage, establishing the interaction effect. In the second plot, shown in panel A.2 in Figure A.1, we plot the effect of degree centrality (x axis) on the predicted value of time to product release (y axis, measured in log-hours) at two levels of brokerage, keeping all the other variables constant at their means. The plot in panel A.2 shows the impact of the interaction of degree centrality and brokerage on time to product release. From panel A.2, time to product release decreases by a maximum of 31% before increasing, as degree centrality changes from low to high under high brokerage, compared with an 11% decrease under low brokerage.

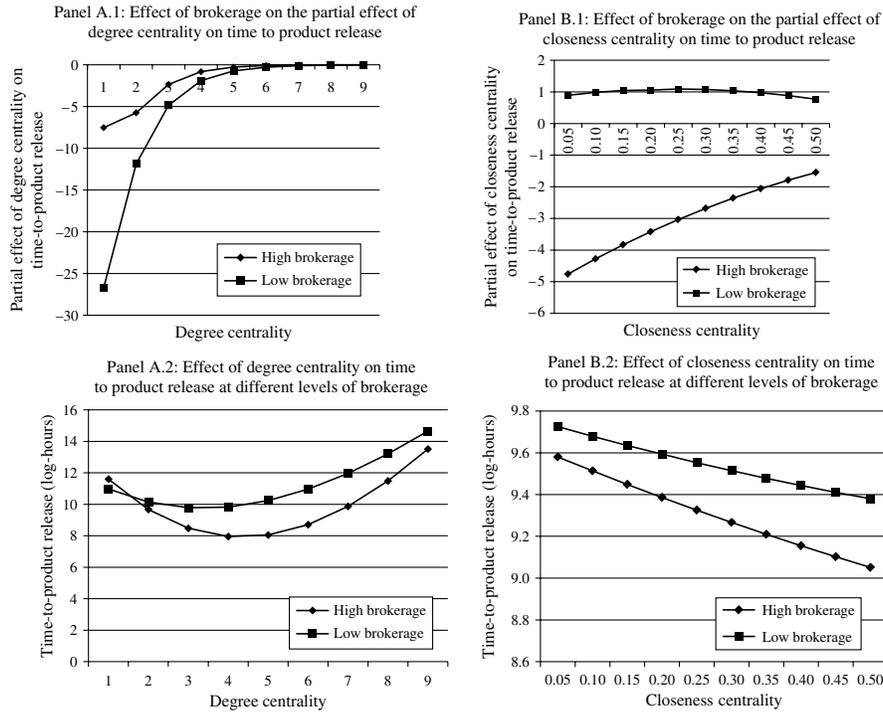
In panel B.1 in Figure A.1, we show how the partial effect varies with closeness centrality (x axis, 0.05–0.5 represents the sample range) across two levels of brokerage (25th versus 75th percentile), fixing all the other variables at their means. The changing distance between the two curves establishes the interaction effect. In panel B.2, we plot the effect of closeness centrality on the predicted value of time to product release for two levels of brokerage. From panel B.2, we see that time to product release decreases by 5.5% as closeness centrality increases under high brokerage compared with a 3.5% decrease under low brokerage.

Interaction of Founder's Social Capital and Degree of User Engagement

In panel A.1 in Figure A.2, we show how the partial effect of degree centrality (x axis, 1–9 represents the sample range) on time to product release varies across two levels of user engagement (25th versus 75th percentile), and in panel A.2, we plot the effect of degree centrality on time to product release across the two levels of user engagement. The changing distance between the curves in panel A.1 establishes the interaction effect, and from panel A.2, we see that time to product release decreases by 83% as degree centrality increases with a high degree of user engagement, compared with a 40% decrease for a low value.

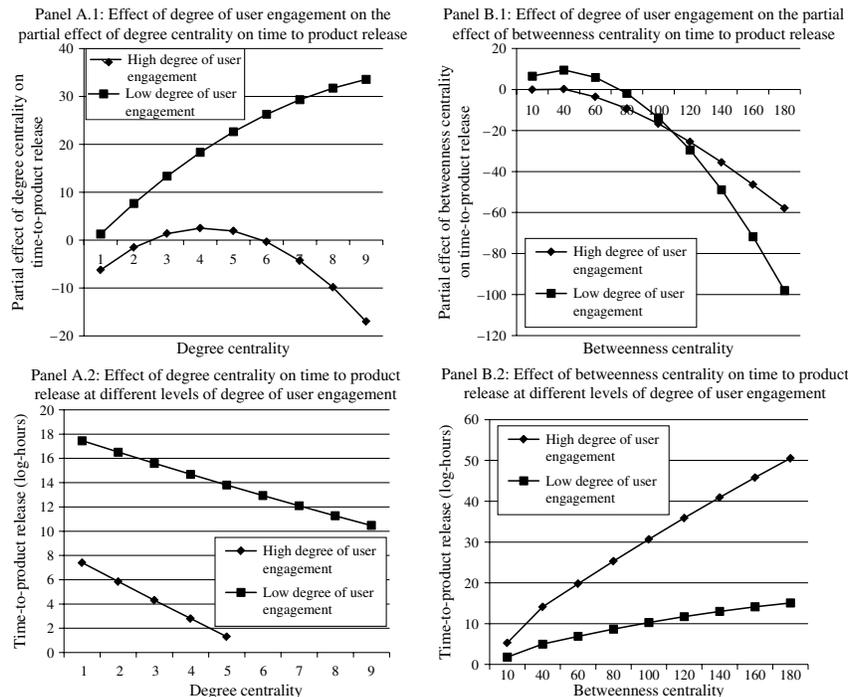
In panel B.1 of Figure A.2, we depict how the partial effect of betweenness centrality varies across two levels of degree of user engagement, and in panel B.2 we plot the effect of betweenness centrality on time to product release across the two levels of degree of user engagement. From panel B.2, we see that time to product release decreases by 10% as closeness centrality increases from low to high for user-focused projects, compared with a 5% decrease for developer-focused projects.

Figure A.1 Plots for the Interaction of Embeddedness and Brokerage



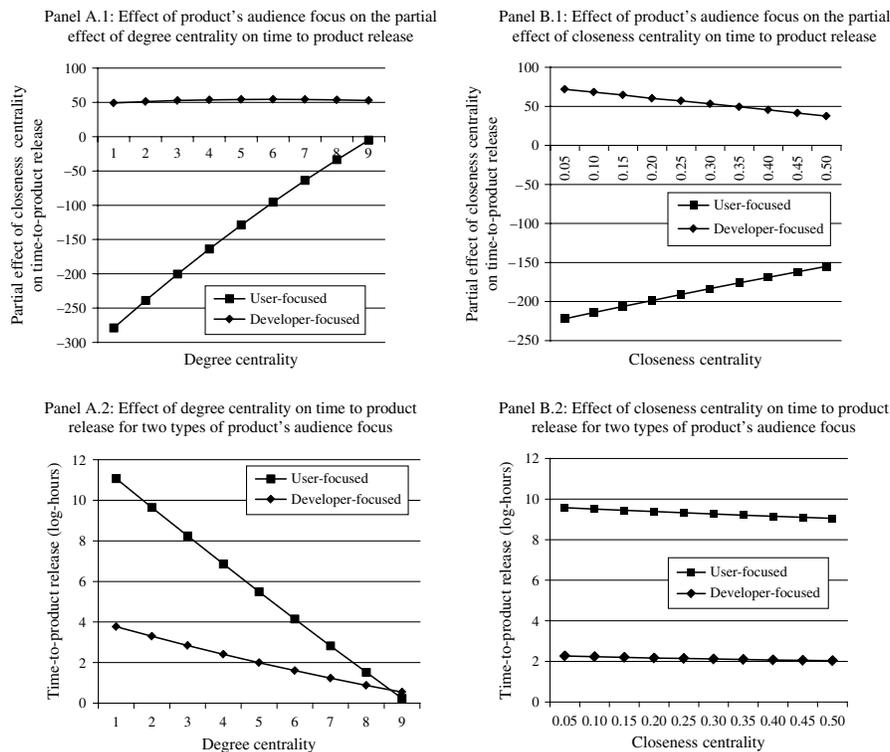
Notes. We used the 25th and the 75th percentile values in the sample for the low and high conditions of brokerage. For degree and closeness centrality, plotted on the x axis, we used their sample range. The partial effect, plotted on the y axis in panels A.1 and B.1, is the unit change in time to product release for a unit change in degree and closeness centrality, respectively. Time to product release, plotted in panels A.2 and B.2, is the predicted value of the dependent variable in log-hours from the estimation.

Figure A.2 Plots for the Interaction of Founder's Social Capital and Degree of User Engagement



Notes. We used the 25th and the 75th percentile values in the sample for the low and high conditions of degree of user engagement. For degree and betweenness centrality, plotted on the x axis, we used their sample range. The partial effect, plotted on the y axis in panels A.1 and B.1, is the unit change in time to product release for a unit change in degree and betweenness centrality, respectively. Time to product release, plotted in panels A.2 and B.2, is the predicted value of the dependent variable in log-hours from the estimation.

Figure A.3 Plots for the Interaction of Founder's Social Capital and Product's Audience Focus



Notes. Product's audience focus is measured by a dichotomous variable: user-focused versus developer-focused. For degree and closeness centrality, plotted on the x axis, we used their sample range. The partial effect, plotted on the y axis in panels A.1 and B.1, is the unit change in time to product release for a unit change in degree and closeness centrality, respectively. Time to product release, plotted in panels A.2 and B.2, is the predicted value of the dependent variable in log-hours from the estimation.

Interaction of Founder's Social Capital and Product's Audience Focus

In panel A.1 in Figure A.3, we see how the partial effect of degree centrality on time to product release varies across the two types of audience focus, and in panel A.2, we plot the effect of degree centrality on the predicted value of time to product release for the two types. The changing distance between the curves in panel A.1 establishes the interaction effect, and panel A.2 depicts the impact of the interaction on the dependent variable. From panel A.2, we see that time to product release decreases by 98% as degree centrality increases from low to high for user-focused projects, compared with an 86% decrease for developer-focused projects.

In panel B.1 of Figure A.3, we see how the partial effect of closeness centrality on time to product release varies across the two types of audience focus, and in panel B.2, we plot the effect of closeness centrality on time to product release for the two types. The changing distance between the curves in panel B.1 establishes the interaction effect, and panel B.2 shows the impact of the interaction on the dependent variable. From panel B.2, time to product release decreases by 10% as closeness centrality increases from low to high for user-focused projects, compared with a 5% decrease for developer-focused projects.

References

Ahuja, G. 2000. Collaboration networks, structural holes, and innovation: A longitudinal study. *Admin. Sci. Quart.* 45(3) 425–455.

- Ai, C., E. C. Norton. 2003. Interaction terms in logit and probit models. *Econom. Lett.* 80(1) 123–129.
- Barczak, G., A. Griffin, K. B. Kahn. 2009. Trends and drivers of success in NPD practices: Results of the 2003 PDMA best practices study. *J. Product Innovation Management* 26(1) 3–23.
- Batagelj, V., A. Mrvar. 1998. Pajek-program for large network analysis. *Connections* 21(2) 47–57.
- Bayus, B. L., S. Jain, A. G. Rao. 1997. Too little, too early: Introduction timing and new product performance in the personal digital assistant industry. *J. Marketing Res.* 34(1) 50–63.
- Bhat, C. R. 2003. Simulation estimation of mixed discrete choice models using randomized and scrambled Halton sequences. *Transportation Res.* 37(9) 837–855.
- Burt, R. S. 1992. *Structural Holes: The Social Structure of Competition*. Harvard University Press, Cambridge, MA.
- Burt, R. S. 1997. The contingent value of social capital. *Admin. Sci. Quart.* 42(2) 339–365.
- Burt, R. S. 2004. Structural holes and good ideas. *Amer. J. Sociol.* 110(2) 349–399.
- Burt, R. S. 2005. *Brokerage and Closure: An Introduction to Social Capital*. Oxford University Press, New York.
- Chandrashekar, M., R. Mehta, R. Chandrashekar, R. Grewal. 1999. Market motives, distinctive capabilities, and domestic inertia: A hybrid model of innovation generation. *J. Marketing Res.* 36(1) 95–112.
- Coleman, J. S. 1988. Social capital in the creation of human capital. *Amer. J. Sociol.* 94(1) 95–120.
- Crowston, K., H. Annabi, J. Howison. 2003. Defining open source software project success. *Proc. Internat. Conf. Inform. Systems*, Association for Information Systems, Atlanta, 327–340.

- Dekimpe, M. G., L. M. Van de Gucht, D. M. Hanssens, K. I. Powers. 1998. Long-run abstinence after narcotics abuse: What are the odds? *Management Sci.* **44**(11) 1478–1492.
- Elfenbein, D. W., T. R. Zenger. 2010. The economics of relational (social) capital: Exploring the value of exchange relationships in industrial procurement. Working paper, Washington University in St. Louis, St. Louis. <http://ssrn.com/abstract=1351852>.
- Erdem, T., M. P. Keane. 1996. Decision-making under uncertainty: Capturing dynamic brand choice processes in turbulent consumer goods markets. *Marketing Sci.* **15**(1) 1–20.
- Fleming, L., D. M. Waguespack. 2007. Brokerage, boundary spanning, and leadership in open innovation communities. *Organ. Sci.* **18**(2) 165–180.
- Fleming, L., S. Mingo, D. Chen. 2007. Collaborative brokerage, generative creativity and creative success. *Admin. Sci. Quart.* **52**(3) 443–475.
- Granovetter, M. S. 1973. The strength of weak ties. *Amer. J. Sociol.* **78**(6) 1360–1380.
- Granovetter, M. S. 1985. Economic action and social structure: The problem of embeddedness. *Amer. J. Sociol.* **91**(3) 481–510.
- Greene, W. H. 2003. *Econometric Analysis*, 5th ed. Prentice-Hall, Upper Saddle River, NJ.
- Greene, W. H. 2010. Testing hypotheses about interaction terms in nonlinear models. *Econom. Lett.* **107**(2) 291–296.
- Grewal, R., G. L. Lilien, G. Mallapragada. 2006. Location, location, location: How network embeddedness affects project success in open source systems. *Management Sci.* **52**(7) 1043–1056.
- Grewal, R., R. Mehta, F. R. Kardes. 2004. The timing of repeat purchases of consumer durable goods: The role of functional bases of consumer attitudes. *J. Marketing Res.* **41**(1) 101–115.
- Hahn, J., J. Y. Moon, C. Zhang. 2008. Emergence of new project teams from open source software developer networks: Impact of prior collaboration ties. *Inform. Systems Res.* **19**(3) 369–391.
- Henard, D. H., D. M. Szymanski. 2001. Why some new products are more successful than others. *J. Marketing Res.* **38**(3) 362–375.
- Hendricks, K. B., V. R. Singhal. 1997. Delays in new product introductions and the market value of the firm: The consequences of being late to the market. *Management Sci.* **43**(4) 422–436.
- Kalbfleisch, J. D., R. L. Prentice. 1980. *The Statistical Analysis of Failure Time Data*. John Wiley & Sons, New York.
- Kiefer, N. M. 1988. Economic duration data and hazard functions. *J. Econom. Literature* **26**(2) 646–679.
- Kogut, B., U. Zander. 1992. Knowledge of the firm, combinative capabilities, and the replication of technology. *Organ. Sci.* **3**(3) 383–397.
- Lakhani, K. R., E. von Hippel. 2002. How open source software works: “Free” user-to-user assistance. *Res. Policy* **32**(6) 923–943.
- Langerak, F., E. J. Hultink. 2006. The impact of product innovativeness on the link between development speed and new product profitability. *J. Product Innovation Management* **23**(3) 203–214.
- Lee, S.-Y. T., H.-W. Kim, S. Gupta. 2009. Measuring open source software success. *Omega* **37**(2) 426–438.
- Lin, N. 2001. Building a network theory of social capital. N. Lin, K. Cook, R. S. Burt, A. De Gruyter, eds. *Social Capital: Theory and Research*. Hawthorne, New York, 3–29.
- Madey, G. 2005. SourceForge.net research data archive. University of Notre Dame, Notre Dame, IN. Retrieved April 2011, <http://www.nd.edu/~oss/Data/data.html>.
- March, J. G. 1991. Exploration and exploitation in organizational learning. *Organ. Sci.* **2**(1) 71–87.
- McFadyen, M. A., A. A. Cannella Jr. 2004. Social capital and knowledge creation: Diminishing returns of the number and strength of exchange relationships. *Acad. Management J.* **47**(5) 735–746.
- Morrison, D. G. 1969. On the interpretation of discriminant analysis. *J. Marketing Res.* **6**(2) 156–163.
- Nerkar, A., S. Paruchuri. 2005. Evolution of R&D capabilities: The role of knowledge networks within a firm. *Management Sci.* **51**(5) 771–785.
- Obstfeld, D. 2005. Social networks, the *tertius iungens* orientation, and involvement in innovation. *Admin. Sci. Quart.* **50**(1) 100–130.
- O’Hern, M. S., A. Rindfleisch, D. A. Schweidel, K. Antia. 2011. The impact of user-generated content on product innovation. Working paper, University of Wisconsin–Madison, Madison. <http://ssrn.com/abstract=1843250>.
- O’Mahony, S., F. Ferraro. 2007. The emergence of governance in an open source community. *Acad. Management J.* **50**(5) 1079–1106.
- Park, S., S. Gupta. 2009. Simulated maximum likelihood estimator for the random coefficient logit model using aggregate data. *J. Marketing Res.* **46**(4) 531–542.
- Raymond, E. 1999. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O’Reilly, Cambridge, MA.
- Reibstein, D. J., G. Day, J. Wind. 2009. Is marketing academia losing its way? *J. Marketing* **73**(4) 1–3.
- Romanelli, E. 1989. Environments and strategies of organization start-up: Effects on early survival. *Admin. Sci. Quart.* **34**(3) 369–387.
- Ronchetto, J. R., M. D. Hutt, P. H. Reingen. 1989. Embedded influence patterns in organizational buying systems. *J. Marketing* **53**(4) 51–62.
- Rosa, J. A., J. F. Porac, J. Runser-Spanjol, M. S. Saxon. 1999. Sociocognitive dynamics in a product market. *J. Marketing* **63**(Special issue) 64–77.
- Sastry, N. 1997. A nested frailty model for survival data, with an application to the study of child survival in northeast Brazil. *J. Amer. Statist. Assoc.* **92** 426–435.
- Schmidt, P., A. D. Witte. 1989. Predicting criminal recidivism using “split population” survival time models. *J. Econometrics* **40**(1) 141–159.
- Sethi, R., D. C. Smith, C. W. Park. 2001. Cross-functional product development teams, creativity, and the innovativeness of new consumer products. *J. Marketing Res.* **38**(1) 73–85.
- Shah, S. K. 2006. Motivation, governance, and the viability of hybrid forms in open source software development. *Management Sci.* **52**(7) 1000–1014.
- Sinha, R. K., M. Chandrashekar. 1992. A split hazard model for analyzing the diffusion of innovations. *J. Marketing Res.* **29**(1) 116–127.
- Smith, P. G. 1999. From experience: Reaping benefit from speed to market. *J. Product Innovation Management* **16**(3) 222–230.
- Song, X. M., M. E. Parry. 1997. A cross-national comparative study of new product development processes: Japan and the United States. *J. Marketing* **61**(2) 1–18.
- Stephen, A. T., O. Toubia. 2010. Deriving value from social commerce networks. *J. Marketing Res.* **47**(2) 215–228.
- Stewart, K. J., A. P. Ammeter, L. M. Maruping. 2006. Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects. *Inform Systems Res.* **17**(2) 126–144.
- Stinchcombe, A. L. 1965. *Organizations and Social Structure*. Oxford University Press, New York.

- Subramaniam, C., R. Sen, M. L. Nelson. 2009. Determinants of open source software project success: A longitudinal study. *Decision Support Systems* 46(2) 576–585.
- Swaminathan, V., C. Moorman. 2009. Marketing alliances, firm networks, and firm value creation. *J. Marketing* 73(5) 52–69.
- Train, K. 2003. *Discrete Choice Methods with Simulation*. Cambridge University Press, Cambridge, UK.
- Uzzi, B. 1997. Social structure and competition in interfirm networks: The paradox of embeddedness. *Admin. Sci. Quart.* 42(1) 35–67.
- Villas-Boas, J. M., R. S. Winer. 1999. Endogeneity in brand choice models. *Management Sci.* 45(10) 1324–1338.
- von Krogh, G., E. von Hippel. 2006. The promise of research on open source software. *Management Sci.* 52(7) 975–983.
- Wasserman, S., K. Faust. 1999. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, UK.
- Zucker, L. G., M. R. Darby, M. B. Brewer, Y. Peng. 1995. *Collaboration Structure and Information Dilemmas in Biotechnology*. Sage, Thousand Oaks, CA.