# Management Science

# Location, Location, Location: How Network Embeddedness Affects Project Success in Open Source Systems

Rajdeep Grewal, Gary L. Lilien, Girish Mallapragada,

MANAGEMENT SCIENCE
Vol. 52, No. 7, July 2006, pp. 1043–1056
ISSN 0025-1909 | EISSN 1526-5501 | 06 | 5207 | 1043

informs®
DOI 10.1287/mnsc.1060.0550
© 2006 INFORMS

# Location, Location, Location: How Network Embeddedness Affects Project Success in Open Source Systems

Rajdeep Grewal, Gary L. Lilien, Girish Mallapragada
Smeal College of Business, Pennsylvania State University, University Park, Pennsylvania 16802-1009
{rug2@psu.edu, glilien@psu.edu, gzm108@psu.edu}

The community-based model for software development in open source environments is becoming a viable alternative to traditional firm-based models. To better understand the workings of open source environments, we examine the effects of network embeddedness—or the nature of the relationship among projects and developers—on the success of open source projects. We find that considerable heterogeneity exists in the network embeddedness of open source projects and project managers. We use a visual representation of the affiliation network of projects and developers as well as a formal statistical analysis to demonstrate this heterogeneity and to investigate how these structures differ across projects and project managers. Our main results surround the effect of this differential network embeddedness on project success. We find that network embeddedness has strong and significant effects on both technical and commercial success, but that those effects are quite complex. We use latent class regression analysis to show that multiple regimes exist and that some of the effects of network embeddedness are positive under some regimes and negative under others. We use project age and number of page views to provide insights into the direction of the effect of network embeddedness on project success. Our findings show that different aspects of network embeddedness have powerful but subtle effects on project success and suggest that this is a rich environment for further study.

*Key words*: network embeddedness; open source software; affiliation network; latent class analysis
*History*: Accepted by Eric von Hippel and Georg von Krogh, guest editors; received September 17, 2004. This paper was with the authors 4 months for 3 revisions.

## 1. Introduction[1]

The open source, community-based model of software development is becoming a viable alternative to the traditional firm-based model. With IBM endorsing Linux as a viable operating system option and contributing its source code for speech recognition and relational database software to various open source initiatives (e.g., Lohr 2004), and with Microsoft explicitly recognizing its competitive rivalry with Linux (e.g., Spencer and Greene 2003), this new model has achieved market legitimacy (e.g., von Hippel 2001, von Hippel and von Krogh 2003). The primary emphasis of open source systems is on developing software such that the source code is public. The level of success of the resulting new code from open source software development projects will likely determine the stature and long-term viability of this community-based movement (e.g., Lakhani and Wolf 2003, von Hippel and von Krogh 2003). The legitimacy of this model of software development provides both an opportunity and some challenges. The opportunity is that this self-generating, collaborative model may provide new templates that can enhance the efficiency and effectiveness of the new product development process. The challenges are to see if (1) there are sufficient differences in the types of collaborative structures that have thus far emerged to infer which models work and which don't, and, if there are (2) to measure and quantify the relationship between these structural differences and the success and failures of the associated software development projects. We report here on research using data on multiple projects collected from a consortium of open source projects, specifically SourceForge.net, to address these two challenges.

The naturally evolving structure of the relationships between the developers involved and the project that they are working on—the social capital involved in the system—provides a critical focus for the distinction of the open source movement from more traditional software development mechanisms. Recognizing the criticality of social capital (e.g., Portes 1998), organizational researchers have highlighted the importance

[1] See the online companion on the *Management Science* website at http://mansci.pubs.inform.org/ecompanion.html for a discussion of the open source movement, SourceForge. net, the methodologies used here, and other supplementary analyses.

of embeddedness—the architectural nature of interfirm relationships—in organizational activities such as receiving financing (e.g., Uzzi 1999), distribution of power in interfirm relationships (Yamagishi et al. 1988), and hiring top managers (Granovetter 1995). Building on the research in organizational sociology, we suggest that social capital and the ensuing network embeddedness (e.g., Granovetter 1985, Uzzi 1996) are likely to have a significant impact on the success of open source software development projects. Thus our research first identifies the nature of network embeddedness in open source systems and then relates this embeddedness to the success of open source projects.

We study a foundry (a related set of projects) and associated projects at SourceForge.net, comprising 108 projects and 490 developers, and find that the organizational structure—network embeddedness—does differ significantly across these projects. After controlling for more standard descriptors such as number of bugs fixed, we find that the degree and nature of network embeddedness of both projects and project managers does indeed influence project success. The pattern of this influence is quite complex, however, in that greater embeddedness is not always beneficial. We use project age, which signals stage of project life cycle, and number of page views, which assesses market potential and project popularity, to provide some insights into when the effect of network embeddedness on project success is positive and when it is negative.

We proceed as follows. In §2, we provide the conceptual background and research hypotheses. There we show why it is appropriate to view open source systems as networks and discuss the relevant literature on embeddedness. With a project as a unit of analysis, in §3, we outline our strategy for data collection and present the results. We first establish heterogeneity in the network embeddedness of projects and project managers, and then relate network embeddedness of projects and project managers to project success. We conclude, in §4, by discussing our findings, providing directions for further research, and elaborating on managerial implications.

## 2. Conceptual Background and Research Hypotheses

We argue that social capital varies across projects and developers and that it plays a critical role in the success of open source projects. We view social capital as the relations among developers, including project managers, and projects that provide developers access to information and (perhaps) embedded resources (e.g., Portes 1998). The analysis of social capital focuses on what is referred to as the network effect (e.g., Ruef et al. 2003) or embeddedness (e.g., Granovetter 1985). Here, we refer to this effect as network embeddedness. The emphasis in this line of investigation is to examine the importance of project managers' (projects') location: how central is that location (e.g., Portes 1998), and how strong are the ties that the location provides (e.g., Granovetter 1973). Central locations with stronger ties increase social capital and network embeddedness. We begin by justifying our use of social networks to study open source systems, and then develop hypotheses related to project success.

### 2.1. Open Source Systems as Networks

Software development in the community-based model of the open source movement involves collaboration among developers working in teams. Often, developers work on multiple software development projects, and thus belong to multiple teams. The importance of teams in new product development is well established and research has demonstrated the critical role of team leaders, the importance of team composition, and the criticality of team chemistry for project success (e.g., Sarin and Mahajan 2001). The structure of software development teams should also be important in the open source environment. These software development teams are largely self-organized, i.e., the hierarchical structure that exists within firms does not directly manifest itself in the community-based model (e.g., Lakhani and Wolf 2003). Social capital, then, seems likely to substitute for the positional power that comes from the hierarchical structure that exists within firms. Specifically, project managers with social capital should find it easier to put together teams with the requisite skill sets, and the projects initiated by these more embedded developers should be more sought after (e.g., Ruef et al. 2003).

### 2.2. Two-Mode Affiliation Networks

To evaluate the presence and the consequences of this heterogeneity for project success, we rely on two-mode affiliation networks (e.g., Faust 1997). In our case, the actors are developers, the events are projects,[2] actors are related to each other through events, and events are related to other events because of common actors. Thus, in our case, developers are related to one another because they work together on projects and projects are related to one another because they share developers (for an example, see the appendix).

[2] Because we use a project as a unit of analysis and because a project may have multiple developers, we assess actor embeddedness by measuring the embeddedness of the project manager.

### 2.3. Forms of Network Embeddedness

In their critique of neoclassical economics and subsequent efforts by economists to relax assumptions of rationality and perfect information (Williamson 1985, North 1990), organizational sociologists argue that organizational routines, processes, and structures are embedded in the broader social context (Smelser and Swedberg 1994). Typically, researchers have proposed four broad categories of embeddedness: cognitive, cultural, structural, and political (Zukin and DiMaggio 1990). According to Zukin and DiMaggio (1990), cognitive embeddedness refers to the "ways in which the structured regularities of mental processes limit the exercise of economic reasoning" (pp. 15–16); cultural embeddedness refers to the "role of shared collective understandings in shaping economic strategies and goals" (p. 17); structural embeddedness refers to the "contextualization of economic exchange in the pattern of ongoing interpersonal relations" (p. 18); and political embeddedness refers to the "manner in which economic institutions and decisions are shaped by a struggle for power that involves economic actors and nonmarket institutions, particularly the state and social classes" (p. 20). The focus of our research is on what Zukin and DiMaggio (1990) refer to as structural embeddedness.

However, as empirical research into this subject is just beginning to emerge, one can find several theoretical variants of structural embeddedness. For example, Uzzi (1996, p. 675) suggests that "structural embeddedness focuses on relational quality of interactor exchanges and the architecture of network ties," thereby subsuming three distinct constructs of Gulati and Gargiulo (1999), i.e., relational, positional, and structural embeddedness. Gulati and Gargiulo (1999, p. 1446) view structural embeddedness more narrowly and define it as "the structure of relationships around actors." In contrast, Gulati (1998, p. 296) uses the terms structural and positional embeddedness interchangeably.

Here, we use the term "network embeddedness" to capture the *architecture of network ties*, and then define three subconstructs to represent network embeddedness, i.e., structural, junctional, and positional embeddedness. Structural embeddedness captures the extent to which an entity is entrenched in a network of relationships, junctional embeddedness assess the extent to which an entity connects other entities, and positional embeddedness appraises the extent to which an entity is connected with other structurally embedded entities. Higher values on any of the three network embeddedness subconstructs would imply greater embeddedness and social capital. The appendix operationalizes these constructs.

### 2.4. Project Success

Unlike traditional firm-driven endeavors, open source projects are not always driven by direct profit motives (e.g., Lakhani and Wolf 2003), and therefore it is not always clear how to define success for such projects. Nonetheless, the criteria for success of open source projects should encompass both the technical achievements of a project, as well as indicators of market or commercial success. This pair of criteria for project success is consistent with the literature in information systems on software success (e.g., Rai et al. 2002) and the literature on R&D success in new product development going back to Mansfield and Wagner (1975). Thus we seek to link network embeddedness to project technical and commercial success.

### 2.5. Research Hypotheses

To understand how heterogeneity in social capital and network embeddedness of projects and project managers influences the success of the projects, we must first establish that heterogeneity does indeed exist in the embeddedness of projects and project managers. A cursory examination of open source projects reveals considerable variation in various aspects of the projects such as (1) the background of project managers (they work for different firms, vary in skill sets, etc.), (2) objectives of the projects (e.g., usage context—database software as opposed to text editor), and (3) scale of the project, which could result in a larger number of developers and longer lifespan of the projects. Thus, consistent with organizational research in other contexts (e.g., Uzzi 1996), we expect to find significant heterogeneity in the embeddedness of projects and project managers.

Hypothesis 1 (H1). *Significant heterogeneity exists in the network embeddedness of open source projects and project managers.*

Assuming that we establish heterogeneity (i.e., find support for H1), we propose four hypotheses, i.e., two on the influence of project embeddedness on technical and commercial success and two on the influence of project manager embeddedness on technical and commercial success.

**2.5.1. Project Network Embeddedness and Technical Project Success.** When project embeddedness is high, projects have access to greater resources because of the larger number of developers (structural embeddedness) and the better information quality because of developers' linkages with other projects in general (junctional embeddedness), and other important projects in particular (positional embeddedness) (e.g., Freeman 1979). Thus, a high degree of network embeddedness implies that the complex tasks associated with software development can be spread over more developers, resulting in better organization, and hence higher productivity. The development process, which involves tasks such as code development, debugging, document writing, translation, and consulting can be better handled with greater resources and should lead to more technical success. Access to higher quality information should also increase the technical success of projects, as it tends to be more relevant, has greater accuracy and reliability, and tends to be timely (e.g., O'Reilly 1982). Research in diverse contexts such as on stock returns (e.g., Veronesi 2000) and decision quality (e.g., Raghunathan 1999) shows that high-quality information is used more frequently and results in better outcomes than does low-quality information (e.g., Maltz and Kohli 1996). Indeed, research in social networks shows that embeddedness is an important indicator of group performance (e.g., Freeman et al. 1980).

Hypothesis 2 (H2). *The network embeddedness of a project will positively influence the technical success of the project.*

**2.5.2. Project Manager Network Embeddedness and Technical Project Success.** A project manager plays the key role of coordinating overall project development activity. Project manager embeddedness is higher when the manager works on more projects (structural embeddedness), serves as a conduit for information exchange among project teams (junctional embeddedness), and participates in important (embedded) projects (positional embeddedness). The larger the number of linkages and the more important the linkages, the higher the project manager's information quality will be, resulting in greater technical success. In contrast, high network embeddedness also implies that the project manager is working on more projects and may be exposed to too much information, leading to cognitive overload and poorer work performance (e.g., Rosa et al. 1999), resulting in lower technical success. Thus the influence of project manager embeddedness on project technical success should be positive for some projects and negative for others. In fact, as projects age, the management of the projects becomes more streamlined because developers better understand their roles and norms of interactions among the developers are well established. Thus, high-quality information should be more useful in newer projects, and the value of project manager embeddedness should decline as projects age.

Hypothesis 3A (H3A). *The network embeddedness of a project manager will positively influence the technical success of the project for some projects and negatively influences the technical success of others.*

Hypothesis 3B (H3B). *The likelihood that project manager embeddedness positively influences project technical success will decline as project age increases.*

**2.5.3. Project Network Embeddedness and Commercial Project Success.** Signaling theory suggests that project network embeddedness signals project quality such that greater embeddedness will imply higher quality, i.e., the users are likely to infer that more connected projects are of higher quality (e.g., Spence 1974). Similarly, if project network embeddedness is a signal of software quality being developed, then it should increase the likelihood of commercial success.

The literature on social networks and diffusion of innovations shows that network structures influence the rate at which innovations diffuse (e.g., Abrahamson and Rosenkopf 1997), suggesting that embedded projects are able to more successfully disseminate project information. Clearly, the effect on success would depend on the valence of the information communicated, positive or negative (e.g., Mahajan et al. 1984), where the valence depends on the reputation of the developers. As in the case of corporate reputation (e.g., Fombrun and Shanley 1990), reputation in the open source environment should be a multidimensional construct. For example, a project manager may have the reputation of developing technically sophisticated (good reputation) software that is not user-friendly (bad reputation). Project network embeddedness would facilitate the dissemination of this information. When the valence of the salient reputation dimension is positive (negative), word of mouth should increase (decrease) the commercial success of the project. Thus, project network embeddedness can have a positive or a negative effect on commercial project success. In our context, the number of page views, which is an indicator of market potential and popularity of the project, should indicate whether the effect of project network embeddedness on commercial success would be positive or negative. When there is positive word of mouth within the network of users, social contagion effects (Van den Bulte and Lilien 2001) would result in more users visiting the project website thereby increasing page views. In contrast, negative word of mouth would dissuade users from visiting the project website, thus lowering page views.

Hypothesis 4A (H4A). *The network embeddedness of a project will positively influence the commercial success of the project for some projects and negatively influences the commercial success of other projects.*

Hypothesis 4B (H4B). *The likelihood that project embeddedness positively influences project commercial success increases as the number of page views increases.*

**2.5.4. Project Manager Network Embeddedness and Commercial Project Success.** If a project manager's network embeddedness signals project quality, then project manager embeddedness should positively influence project success (e.g., Spence 1974). Project manager network embeddedness should also facilitate the dissemination of word-of-mouth information concerning the project (e.g., Deroian 2002). Again, the valence of information disseminated, which would depend on the reputation of project and its developers, would determine whether commercial success is enhanced or reduced. Yet, again, because of social contagion effects, positive word of mouth within the network of users would result in more users visiting the project website, thereby increasing page views and negative word of mouth would dissuade users from visiting the project website, thus lowering page views (Van den Bulte and Lilien 2001). Thus, parallel to the previous hypothesis, we suggest:

Hypothesis 5A (H5A). *The network embeddedness of a project manager will positively influence the commercial success of the project for some projects and negatively influences the commercial success of others.*

Hypothesis 5B (H5B). *The likelihood that project manager embeddedness positively influences project commercial success will increase as the number of page views increases.*

## 3. The Study

### 3.1. Data Source and Data Collection Procedure

Based on the suggestions of von Hippel and von Krogh 2003, we collect our data from the website SourceForge.net, which is an open source initiative that provides Web space to organize and coordinate open source product development. As of November 2005, the site hosts more than 104,000 projects with more than 1,159,800 registered users. The projects on SourceForge.net are classified under broad technology platforms called *project foundries*. To keep the data collection manageable, we sought a foundry with 8–15 active projects. We randomly selected the "Perl" Foundry, comprising projects that share the Perl programming language as the platform technology. The foundry has 10 active projects that represent a wide range of applications such as databases, system administration, text processing, and development tools. These projects have 72 members, resulting in an affiliation matrix of 72 rows (developers) and 10 columns (projects), where each entry is a 1 if a developer worked on a project and 0 otherwise.

To view this foundry in the framework of the more complete project-developer network, we listed all non-Perl projects that these 72 developers were members of, resulting in 108 projects, including the 10 projects in the Perl Foundry. We also identified all other developers aside from the 72 Perl developers who were members of these additional 98 projects, resulting in a total of 490 developers, including the 72 Perl developers. The resulting sociomatrix has 490 rows (developers) and 108 columns (projects), providing an appropriate sample of projects to represent the Perl affiliation networks (e.g., Faust 1997). The procedure we use for developing the sample is referred to as the nominalist approach (Laumann et al. 1989) and is frequently applied in related research studies (e.g., Granovetter 1995, Wasserman and Faust 1999).

### 3.2. Measures

**3.2.1. Network Embeddedness.** To capture the network embeddedness of projects and developers, we use the notion of centrality that captures the "importance" or "visibility" of projects and developers (e.g., Faust 1997, Freeman 1979). Specifically, we use degree centrality—the number of projects in which the manager participates—to operationalize structural embeddedness, betweenness centrality—the number of paths between other nodes on which the manager lies—to operationalize junctional embeddedness, and eigenvector centrality—the manager participates in important projects—to operationalize positional embeddedness. In a similar manner, one can define centrality for projects. Note that our measure for positional embeddedness has been used by Gulati and Gargiulo (1999), and, consistent with literature on centrality (Wasserman and Faust 1999), we use a centrality-based measure of structural embeddedness (see the appendix for details).

**3.2.2. Project Success Measures.** Software development teams use the Concurrent Versioning System (CVS) to manage the software development process. CVS enables teams to store source code at a central location, thus enabling team members to retrieve the source code to make changes. CVS also helps the team to keep track of every change, including what was changed, when it was changed, and who made the change, and helps in blending changes made by different developers, including ensuring that developers do not accidentally overwrite each others' alterations. A *commit* occurs when a developer uploads the altered source code file, where the CVS tool updates the changed files automatically. As CVS commits reflect meaningful changes to the source code, we treat the number of CVS commits as an indicator of successful technical refinement.

To assess commercial and economic success, we use the number of downloads (DOWN) over the life span of a project. The number of downloads is a market-based measure of popularity, which should relate to product use, particularly when software is distributed through a single channel as in the case of SourceForge.net (e.g., Crowston et al. 2003). When a software product is freely available, researchers have used downloads as a surrogate for "sales" (e.g., Chandrashekaran et al. 1999).

**3.2.3. Other Measures and Covariates.** All of the "new products" that emerge from the Perl Foundry are in the same general market; hence, most of the differentiators of new product success that Cooper (2001) has identified are likely to be common across these projects. There are differences, however, in the age of the project, its market potential or interest level, and the role that users, lead users, in particular, play; factors that we can measure. Number of page views (VIEWS) directly signals the general interest level in the project and its market potential. Because the number of CVS commits (CVS) and number of downloads (DOWN) are likely to increase with the age of a project, we use number of months since the inception of the project (AGE) to control for the age of the project. Users often play a critical role in the development of new products, in general, with lead users being particularly effective in driving success (e.g., von Hippel 2005). The number of bugs closed (BUGS) and support requests (SUPPORT) represent user and lead user input in the open source world, with those requests often having directed solutions associated with them. As discussed earlier, we control for these variables, where we include the counts of bugs closed and support requests answered as correlates and project age (in months), and number of page views as concomitant profiling variables (which we discuss later).

### 3.3. Analysis Approach

Because of qualitative differences between H1 and the other four hypotheses, the statistical approach used to test these hypotheses also varies. We first delineate the approach for testing H1 and then address the approach for testing the others.

**3.3.1. Heterogeneity in Network Embeddedness.** To better understand the nature of network embeddedness in the open source environment (i.e., test H1), we rely on two approaches: (1) a visual approach relying on sociometrics to develop a rich, in-depth description of the relationships among projects and developers (Wasserman and Faust 1999) and (2) a statistical approach based on latent class cluster analysis to formally assess the number of groupings of project structures (Wedel and Kamakura 2000).

**3.3.2. Network Embeddedness and Project Success.** Although both our dependent measures, i.e., CVS and DOWN, are count measures; their mean and standard deviations are fairly large and heavily skewed. Therefore we took the logarithm of these two variables and approximate them as continuous variables. To evaluate the distribution of these two variables, we developed kernel density plots that showed a bimodal distribution, indicating multiple regimes or multiple relationships between each dependent and the independent variables. Latent class regression analysis (e.g., Wedel and Kamakura 2000), which is based on finite mixture theory (e.g., Titterington et al. 1985), provides an appropriate methodology to simultaneously estimate multiple relationships among dependent and independent variable. Specifically for $R$ possible regimes, we specify these relationships as

$$Y_p = \sum_{r=1}^{R} [X_p \beta_r + \varepsilon_r], \tag{1}$$

where $p$ denotes the projects, and $\beta_r$ is the regime-specific regression coefficient. To estimate this multiregime model, we use a finite mixture of linear regressions (DeSarbo and Cron 1988, Wedel and Kamakura 2000), drawing on finite mixture distribution theory (e.g., Titterington et al. 1985). We use Bayes rule to calculate the posterior probability for regime $r$ to be representative of project $p$; that is,

$$P[p \in r \mid Y_p] = \frac{\delta_{r|p} L_{p|r}}{\sum_{r=1}^{R} \delta_{r|p} L_{p|r}}, \tag{2}$$

where $\delta_{r|p}$ denotes the prior probability that project $p$ belongs to regime $r$ and $L_{p|r}$ is the likelihood value that the project $p$ belongs to regime $r$. Consistent with the literature (Dayton and MacReady 1988, Gupta and Chintagunta 1994), we use the logit formulation to specify the prior probabilities as

$$\delta_{r|p} = \frac{e^{\kappa_r}}{\sum_{r=1}^{R} e^{\kappa_r}}, \tag{3}$$

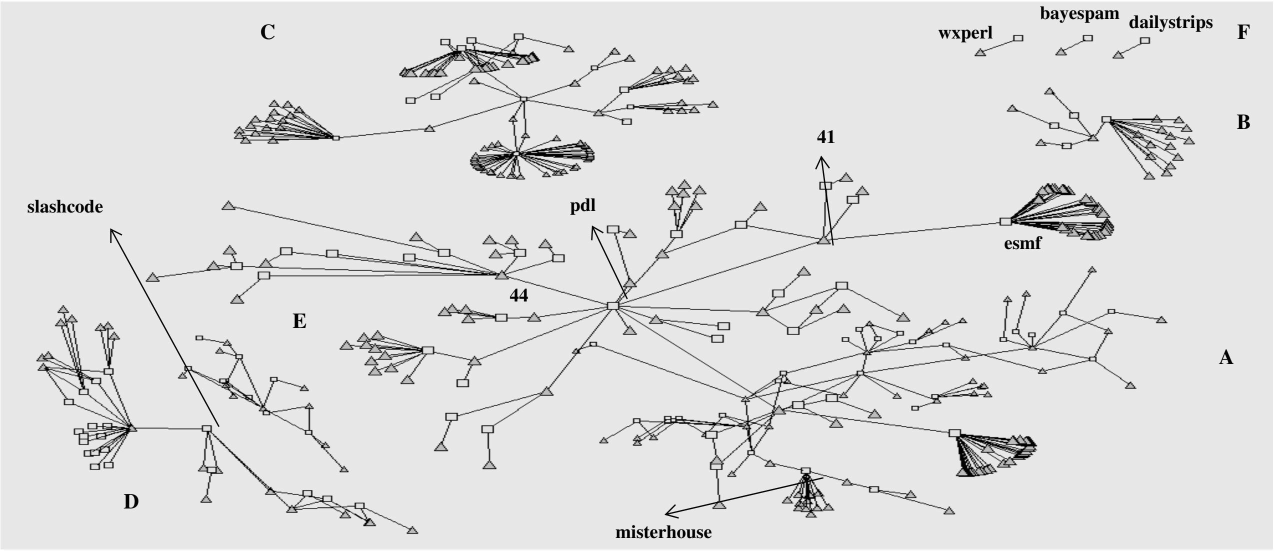
where we estimate $\kappa_r$ for each regime. Again, we can standardize Equation (3) by assuming that $\kappa_R = 1$ (e.g., Gupta and Chintagunta 1994). Thus we treat the last group as the base and only need to estimate $R-1$ parameters. Further, as we hypothesize moderating effects for project age and number of page views, we use the concomitant profiling variable approach to assess the impact of the moderating variables (e.g., Dayton and MacReady 1988). Thus we specify $\kappa_r$ as

$$\kappa_r = \gamma_{0r} + \gamma_{1r} PAGE + \gamma_{2r} PVIEWS, \tag{4}$$

where $\gamma_{0r}$ is the constant, $\gamma_{1r}$ is the effect of project age (PAGE) on the likelihood of belonging to regime $r$, and $\gamma_{2r}$ is the effect of number of page views (PVIEWS) on the likelihood of belonging to regime $r$. The likelihood for each regime is specified based on the standard normal density as $L_{p|r} = \phi^*(\varepsilon_r)$, where, $\phi^*(.)$ is the standardized normal density function and $\varepsilon_r$ is residual error such that $\varepsilon_r \sim N(0, \sigma_r)$. Thus the likelihood function can be written as

$$L = \prod_{p=1}^{P} \sum_{r=1}^{R} \delta_{r|p} L_{p|r}, \tag{5}$$

**Figure 1** Bipartite Graph of the "Perl Foundry Network"



*Notes.* Key features

- The graph is not fully connected with five major clusters (A–E) with Cluster A being the largest and a sixth cluster (Cluster F) of three independent projects ("wxperl," "bayespam," and "dailystrips"), which are from the Perl Foundry.
- Some observations:
  - Developer number 41 in Cluster A works on the largest Perl Foundry project "pdl" with 21 developers and seems to be strategically positioned, as she or he serves as a link for project "esmf" (that has 40 developers, including Developer 41).
  - Developer 44 also works on a project the Perl Foundry "pdl," which seems to have a strategic position.
  - The second largest Perl Foundry project "misterhouse" with 14 developers also belongs in Cluster A, while "slashcode"—a Perl Foundry project with 9 developers—is in Cluster D.

where we have $P$ projects in our data set and estimate the relationship for $R$ regimes. We maximize the natural logarithm of Equation (5) to obtain parameter estimates for an $R$ regime solution. Specifically, we use the E–M algorithm with 50 random starting values to obtain the parameter estimates and determine the number of regimes using the Bayesian Information Criterion (BIC) and the Consistent Akaike Information Criterion (CAIC).[3]

## 3.4. Results

### 3.4.1. Heterogeneity in Network Embeddedness

*3.4.1.1. Visual Representation of Network Structure.* We used the Fruchterman and Reingold (1991) algorithm in the network software package Pajek 1.00 to develop the Perl developer membership bipartite graph. We use squares to represent the projects and triangles to represent the developers (see Figure 1). Note that the Perl Foundry Network in Figure 1 is not fully connected, i.e., there are six clusters (labeled A–F) of projects and developers that do not have connections to other clusters of projects and developers. Cluster A represents the largest connected part of the graph, while Cluster F consists of three projects ("wxperl," "bayespam," and "dailystrips") from the Perl Foundry that have one developer each and do not seem to have a connection with the rest of the network. Developer 41 in Cluster A is strategically positioned and serves as a link for project "esmf" (which has 40 developers, including Developer 41). Developer 41 also works on project "pdl," which is the largest project from the Perl Foundry with 21 developers. Similar to Developer 41, Developer 44 working on project "pdl" has a strategic position. The second largest Perl Foundry project, "misterhouse," with 14 developers, also belongs in Cluster A. In contrast, "slashcode," a Perl Foundry project with nine developers, is in Cluster D. Indeed, Figure 1 strongly suggests that considerable heterogeneity exists in the embeddedness of project and developers in an open source environment.

*3.4.1.2. Latent Class Cluster Analysis.* To formally affirm the visual demonstration of heterogeneity suggested in Figure 1, we seek to establish statistical differences using latent class cluster analysis. We use

[3] Specifically, $BIC = -2*LL + K*\ln(N)$ and $CAIC = -2*LL + K*(1+\ln(N))$, where $LL$, $K$, and $N$ stand for log-likelihood value, number of parameters, and sample size, respectively. We also report an entropy measure of separation (ES) to assess the extent of separation of the clusters (Wedel and Kamakura 2000). ES is bounded in the range 0–1 such that a value closer to 1 indicates good separation of groups or latent clusters, where $ES = 1-[\sum_{n=1}^{N}\sum_{c=1}^{C} -p_{n|c}\ln(p_{n|c})]/[N\ln(C)]$ and $p_{n|c}$ is the probability of unit $n$ belonging to cluster $c$, which we calculate using Bayes rule.

the network embeddedness measures discussed earlier and the likelihood dominance criterion (Pollak and Wales 1991) to show that a model with either two clusters or six clusters (our optimal solution) is superior to a model with a single cluster ($p < 0.01$). A test of one cluster versus more than one cluster is an appropriate test of homogeneity (one cluster) versus more than one cluster (heterogeneity), providing support for H1.

### 3.4.2. Network Embeddedness and Project Success

*3.4.2.1. Model Selection.* The information criteria (BIC and CAIC) suggest that a two-regime model is appropriate for number of CVS commits (Entropy of separation (ES) $= 0.99$).[4] The results are a bit ambiguous for number of downloads with BIC suggesting three regimes and CAIC suggesting two regimes. Given the bimodality in the kernel density plot, we pursued the two-regime solution, and the high entropy of separation for the two-regime solution ($ES = 0.99$) provides further support for this two-regime solution. Thus we explored two-regime solutions for both CVS commits and downloads.

*3.4.2.2. Hypothesis Testing*

*CVS commits (technical success).* In Table 1, we present the results for the two-regime solution for CVS and DOWN models. In H2, we had suggested that project network embeddedness should positively affect project technical success. Our results provide some support for this hypothesis. For structural embeddedness, we find a positive and statistically significant coefficient in Regime 1 ($b = 2.660$, $p < 0.01$), but a statistically nonsignificant coefficient in Regime 2 ($b = 0.207$, $p > 38$). For junctional embeddedness, we find the coefficient to be positive and statistically significant in Regime 2 ($b = 2.517$, $p < 0.01$), but statistically nonsignificant in Regime 1 ($b = -0.214$, $p > 0.84$). For positional embeddedness, the hypothesis is also supported in Regime 2 ($b = 1.977$, $p < 0.01$), but the effect is negative in Regime 1 ($b = -0.568$, $p < 0.10$). Thus, for each of the three embeddedness constructs, we find a positive effect in at least one regime, but for positional embeddedness, we also find a negative effect, although significant at only the 10% level. In Regime 1 (older, popular projects), structural embeddedness seems critical but positional embeddedness hurts, while in Regime 2 (younger, relatively less popular project), junctional and positional embeddedness seems to help.

In H3, we suggested that project manager embeddedness would have a positive effect on project technical success for some projects and a negative effect for other projects (H3A), and that the likelihood that project manager embeddedness positively influences project technical success declines as projects age (H3B). Our results show that the effect of project manager network embeddedness is sometimes positive and sometimes negative, thereby supporting H3A. For structural embeddedness, we find the effect to be negative in Regime 1 ($b = -0.375$, $p < 0.05$) and positive in Regime 2 ($b = 0.266$, $p < 0.10$). For junctional embeddedness, the effect is positive in both regimes (Regime 1: $b = 0.364$, $p < 0.05$; Regime 2: $b = 0.259$, $p < 0.10$); and for positional embeddedness, the effect is statistically nonsignificant in Regime 1 ($b = -0.261$, $p > 0.18$) and negative in Regime 2 ($b = -0.401$, $p < 0.10$). Further, younger projects are more likely to belong to Regime 2 ($b = 0.309$, $p < 0.01$), thereby lending support to H3B. Thus, for younger projects, structural and junctional embeddedness have a positive influence but positional embeddedness has a negative effect, while for older projects, structural embeddedness has a negative effect and junctional embeddedness has a positive influence. The results seem to be more complex than we had envisioned. Specifically, the effect varies between the regimes and across the three embeddedness subconstructs. Therefore, theoretically focusing on each of the three subconstructs separately becomes critical.

In terms of the control variables, the results show that they have statistically significant effects only in Regime 2, where number of CVS commits increases as number of bugs closed decreases ($b = -5.765$, $p < 0.01$), and number of support requests answered increases ($b = 0.377$, $p < 0.05$). In terms of descriptive statistics, we find that when compared with Regime 1, Regime 2 has (1) more downloads, (2) more page views, (3) more bugs, (4) fewer support requests answered, and (5) greater positional embeddedness for projects and project managers.

*Downloads (commercial success).* In H4, we had suggested that the network embeddedness of a project will positively influence the commercial success of the project for some projects and negatively influence the commercial success of others (H4A), and that the likelihood that project embeddedness positively

[4] We sought an $R^2$-type measure of the fit of the latent class model with respect to the aggregate model (one regime). To do this, we computed overall and segment specific $R^2$ values for the latent class based on the mean square error (MSE). Segment-specific MSE is calculated from the difference between observed $Y$ and predicted $Y$, i.e., $E(Y \mid X)$. The posterior segment membership probabilities quantify the contribution of a specific case to the error in that segment. Similarly, we calculated the overall $R^2$ for the latent class model based on MSE (note that segment memberships do not enter the equation in assessing the overall $R^2$). In the model for CVS commits the $R^2$ value for single-regime model is 0.44, while that for the optimal two-regime model is 0.86 for overall model, and 0.55 and 0.93 in the two regimes, respectively. For a number of downloads, the aggregate model gives an $R^2$ value of 0.39, while the two-regime solution has an overall $R^2$ value of 0.88, and 0.38 and 0.99 in the two regimes, respectively.

**Table 1 Latent Class Regression Analysis Results**

| | | | Number of CVS commits (CVS) | | Number of downloads (DOWN) | |
|---|---|---|---|---|---|---|
| | | | Regime 1 | Regime 2 | Regime 1 | Regime 2 |
| Variable type | Variable name | Measure | Estimate (SE) | Estimate (SE) | Estimate (SE) | Estimate (SE) |
| CVS | CVS | Number of CVS commits | — | — | 0.367<br>(1.004) | −0.176<br>(0.242) |
| Control variables | BUGS | Number of bugs closed | 0.187<br>(0.165) | −5.765***<br>(2.321) | 10.127***<br>(2.459) | 1.161***<br>(0.049) |
| | SUPPORT | Number of support requests answered | 0.148<br>(0.187) | 0.377**<br>(0.222) | −1.614***<br>(0.433) | 30.127***<br>(0.864) |
| Project embeddedness | Structural | Degree centrality | 2.66***<br>(0.469) | 0.207<br>(0.281) | 0.111<br>(0.481) | 0.032<br>(0.126) |
| | Junctional | Betweenness centrality | −0.214<br>(0.221) | 2.517***<br>(0.539) | 1.306**<br>(0.774) | −0.275***<br>(0.074) |
| | Positional | Eigenvector centrality | −0.568*<br>(0.356) | 1.977***<br>(0.637) | −1.995*<br>(1.252) | −0.030<br>(0.283) |
| Project manager embeddedness | Structural | Degree centrality | −0.375**<br>(0.207) | 0.266*<br>(0.204) | −0.234<br>(0.280) | −0.012<br>(0.081) |
| | Junctional | Betweenness centrality | 0.364**<br>(0.192) | 0.259*<br>(0.175) | 0.470<br>(0.463) | 0.054<br>(0.059) |
| | Positional | Eigenvector centrality | −0.261<br>(0.292) | −0.401*<br>(0.281) | −1.103***<br>(0.524) | −0.018<br>(0.117) |
| Profiling variables | | | | | | |
| Constant | | | 0.793***<br>(0.342) | — | 0.221*<br>(0.160) | — |
| Concomitant variables | Maturity | Project age (AGE) | 0.309***<br>(0.119) | — | −0.079<br>(0.101) | — |
| | Potential | Number of page views (VIEWS) | 3.056*<br>(1.872) | — | 1.153*<br>(0.812) | — |
| Regime size (%) | | | 60 (55.56) | 48 (44.44) | 57 (52.78) | 51 (47.22) |

*Notes.* We report one-tail tests for statistical significance. For each regime, we have two columns of results. In the first column, we report the regression coefficient and its standard error in parenthesis, and in the second column, we report the mean of the explanatory variable with its standard deviation in parenthesis.

$^{*}p < 0.10$, $^{**}p < 0.05$, $^{***}p < 0.01$.

influences project commercial success as the number of page views increase (H4B). The results show that projects with more page views are more likely to belong to Regime 1 ($b = 1.153$, $p < 0.10$). In Regime 1, junctional embeddedness has a positive effect on commercial success ($b = 1.306$, $p < 0.05$), positional embeddedness has a negative effect on project success ($b = -1.995$, $p < 0.10$), and structural embeddedness does not have a statistically significant effect ($b = 0.111$, $p > 0.59$). In Regime 2, the results are statistically nonsignificant for structural ($b = 0.032$, $p > 0.60$) and positional ($b = -0.030$, $p > 0.45$) embeddedness and are negative for junctional embeddedness ($b = -0.275$, $p < 0.01$). Consistent with our reasoning, these results suggest that network embeddedness is more critical for projects with more page views (supporting H4B), but contrary to what we expected, the results also suggest that the influence of network embeddedness need not be positive for all popular projects. Here, the results also vary across the three embeddedness subconstructs, again highlighting the criticality of theory development at the subconstruct level.

In H5, we had suggested that the network embeddedness of a project manager will positively influence the commercial success of the project for some projects and negatively influences the commercial success of others (H5A), and that the likelihood that project manager embeddedness positively influences project commercial success increases as the number of page views increases (H5B). The results do not support H5. It seems that structural (Regime 1: $b = -0.234$, $p > 0.20$; Regime 2: $b = -0.012$, $p > 0.44$) and junctional ($b = 0.470$, $p > 0.84$; Regime 2: $b = 0.054$, $p > 0.81$) embeddedness do not influence commercial project success and positional embeddedness has a negative effect in Regime 1 ($b = -1.103$, $p < 0.01$), and a statistically nonsignificant effect in Regime 2 ($b = -0.018$, $p > 0.44$). Overall, it seems that network embeddedness is more critical for technical project success than for commercial project success, and that project embeddedness is more critical than project manager embeddedness for both measures of success.

**Table 2 Cross-Classification of Regimes Across Downloads and CVS Commits Analyses**

| | CVS | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Regime 1 | | | | Regime 2 | | | |
| | Regime statistics | Number of developers | CVS commits | Downloads | Regime statistics | Number of developers | CVS commits | Downloads |
| Downloads | Cell A<br>Number of cases: 39 | | | | Cell B<br>Number of cases: 18 | | | |
| Regime 1 | Mean | 5.769 | 1,219.538 | 11,448.385 | Mean | 6.833 | 749.167 | 15,235.889 |
| | SD | 6.831 | 2,228.750 | 22,106.155 | SD | 9.294 | 1,645.448 | 41,974.241 |
| | Exemplars:<br>*dailystrips*<br>*guido*<br>*misterhouse*<br>*wxperl* | | | | Exemplars:<br>*amphetadesk*<br>*bayespam*<br>*spamassassin*<br>ptxdist | | | |
| | Cell C<br>Number of cases: 21 | | | | Cell D<br>Number of cases: 30 | | | |
| Regime 2 | Mean | 6.048 | 569.952 | 2,787.048 | Mean | 4.867 | 2,321.133 | 11,458.733 |
| | SD | 8.102 | 938.985 | 10,251.567 | SD | 8.148 | 11,482.824 | 34,582.479 |
| | Exemplars:<br>*pdl*<br>arboretum<br>hivemind<br>sbeating | | | | Exemplars:<br>*apachetoolbox*<br>*slashcode*<br>iado<br>toolbox | | | |

*Notes.* For each cell, we have provided the overall mean and standard deviation along with a few exemplar projects. The Perl Foundry projects are shown in italic.

Observations

- Cell B has the highest number of downloads.
- Cell D has the highest number of CVS commits.
- Cell A has the second highest number of CVS commits.
- Cell D has the second highest number of downloads.
- Cell C has just one Perl Foundry project. It has the lowest number of both CVS commits and downloads.

In terms of the control variables, we find that technical project success does not impact commercial project success in either regime (Regime 1: $b = 0.367, p > 0.64$; Regime 2: $b = -0.176$, $p > 0.23$). The number of bugs closed does seem to increase the commercial success of projects across both regimes (Regime 1: $b = 10.127$, $p < 0.01$; Regime 2: $b = 1.161$, $p < 0.01$). The number of support requests answered seems to increase project commercial success in Regime 2 ($b = 30.127$, $p < 0.01$) and decrease project commercial success in Regime 1 ($b = -1.614$, $p < 0.01$).

*Comparing regimes.* For both the dependent variables, CVS and DOWN, we found a two-regime solution. One might assert that these two regimes should contain the same projects, i.e., that regime identity should hold across both CVS commits and downloads, a constraint we did *not* impose on the models, which were calibrated independently. To explore this issue, we show these cross-tabulation results in Table 2, which strongly suggest different drivers for regime membership for CVS commits and downloads. The cell sizes range from 18 to 39 and there is no statistically significant difference among them in terms of number of developers. Table 2 provides some commentary on the characteristics of these cells, with Cell B (Regime 1 for Downloads and Regime 2 for CVS commits) highest on average downloads.

## 4. Discussion

We have studied how network embeddedness of projects and developers relate to the success of open source projects. We focused on both technical success, viewed in terms of the number of CVS commits, and commercial success operationalized as the number of downloads. We also suggested that the effects of network embeddedness on technical project success would vary with project age and that commercial project success would vary with the number of page views, which can be seen as an indicator of project market potential and/or popularity. The results generally support the assertion that project network embeddedness positively influences project technical success, while the effect of project manager network embeddedness is more complex and different for older projects when compared with younger projects. The results also suggest that project commercial success is influenced by project network embeddedness and that this influence varies with the number of page views. Overall, the results for the effects of embeddedness are much stronger for technical success than for commercial success, implying that network embeddedness has a greater role to play in technical success than in commercial success. The cause for this greater role may be because embeddedness enables projects to attract talented developers, but is invisible to the users who drive commercial success. In fact, we find no statistically significant link between technical project success and commercial project success.

We must stress the exploratory nature of our research. As research on open systems environments is new, theoretical insights in this domain are just emerging (von Hippel and von Krogh 2003). In this research, we find that significant heterogeneity exists in the embeddedness of open source projects, and there seems no reason to expect this result not to hold for other open source projects. We also find that the architecture of projects and project managers strongly affects technical and commercial project success, a result that should encourage further research in the area.

From a theoretical standpoint, our results suggest several directions for theory development on the effect of network embeddedness on project success. First, it is important to recognize that the effect of network embeddedness varies with the dependent variable, i.e., technical or commercial project success. This finding is consistent with our theoretical development and researchers in this domain could explore those differences more deeply. Second, somewhat contrary to the literature and our assertions, in our empirical analysis, we did not find that the three network embeddedness subconstructs (i.e., structural, junctional, and positional embeddedness) to behave in unison in terms of their effect on project success. For example, project manager's positional embeddedness has a negative effect on technical project success, whereas junctional embeddedness has a positive effect on technical project success. We believe that these differences are likely to be real and research efforts focused on providing theoretical explanations for such differences would be fruitful. Thus, project manager's positional embeddedness, which represents the degree to which the manager is part of the development team of other important projects, could lead to lower technical success of the project because participation in several important projects might result in cognitive overload, and as a result, lower technical performance. Theoretical efforts to develop such ideas would further enrich the understanding of the role of social capital in community-oriented knowledge development systems such as the open source system for software development. Third, our work suggests that it is important to understand the manner in which the effect of network embeddedness subconstructs on project success varies across regimes and to explain those differences. Our research takes some important steps in this direction and we hope that multiregime models are further explored in future research.

Our research has implications for project managers and developers in open source environments and for managers of firms, such as IBM and Sun Microsystems, which are actively participating in open source software projects. For example, assume an executive at IBM is faced with a decision to sponsor projects—either monetarily or by allocation the firm's human resources, or both. Thus, a new product (software) development executive at IBM has to decide which projects IBM programmers work on. The focus of the executive could be on developing a technologically sophisticated product (i.e., focus on technical success) or a commercially viable product, or both. Our results show that projects with more developers see greater technical success in the later stages of project development, i.e., as the projects age. Thus the executive who wants technically superior software would be advised to have larger software development teams and be patient, as after initial habitualization of team norms, the team would have a greater likelihood of technical success (as shown by the coefficient for project structural embeddedness). However, the executive should be aware that if the project leader works on several projects, the technical success of the projects with large teams can be jeopardized (as shown by the coefficient for project manager structural embeddedness).

In general, executives at companies such as IBM should note that (1) project embeddedness is more critical than project manager embeddedness, implying that new managers can reap the benefits of embeddedness if they structure their project teams with care and (2) network embeddedness impacts technical success of the project more than commercial success, and thus executives should focus on network embeddedness when technical achievement is more critical than commercial gains.

Our research has limitations that provide avenues for further research. Besides simple replications of our research, enriched perhaps by more direct observation (via diary, survey, or the like), future research should examine other measures of embeddedness, such as those related to resources and of performance such as rate of innovation in projects and the nature of the innovations (e.g., radical versus incremental). Building in dynamics by examining the effect of structural embeddedness over time should also provide new insights; we have studied this process via a static view, while the dynamics of the network and the environment may have even more powerful effects. For that purpose, one could rely on evolutionary theories in economics or sociology or both. It is our hope that our initial results encourage researchers studying open source systems to embrace a social capital perspective, and that researchers in diverse social sciences will focus on this domain to provide richer insights into open source systems.

An online supplement to this paper is available on the *Management Science* website (http://mansci.pubs.informs.org/ecompanion.html).

## Acknowledgments

The authors contributed equally to this research and are listed alphabetically. The article benefited from the feedback of Bill Ross and Raji Srinivasan and the support of Penn State's Institute for the Study of Business Markets. The authors thank the *Management Science* referees and the special issue editors for their several helpful suggestions.

## Appendix. Two-Mode Affiliation Networks

Consider an affiliation network **A** in which the rows represent the actors (project managers) and the columns represent the events (projects), with 1 when an actor belongs to an event and 0 otherwise. From this nonvalued (i.e., the elements of the matrix are either 0 or 1) affiliation matrix, we can obtain the valued matrix (where higher values indicate greater strength of relationship) for actors ($X^A$) and events ($X^E$) as

$$X^A = AA' \tag{A1a}$$

$$X^E = A'A. \tag{A1b}$$

Thus, for the illustrative example represented in Figure A1, the affiliation matrix $A$ will be

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and its transpose}$$

$$A' = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

And therefore

$$X^A = AA' = \begin{bmatrix} 2 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad \text{and}$$

$$X^E = A'A = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

We define *degree centrality* (operationalizing structural embeddedness) for actor $i$ as ($C_D(X_i^A)$) as (e.g., Faust 1997)

$$C_D(X_i^A) = X_{ii}^A, \tag{A2}$$

where the network has $I$ actors. Thus the degree centrality for an actor $i$ is given by the $i$th diagonal element of $X^A$. The degree centrality for events is calculated in a similar manner.

For information presented in Figure A1, the degree centrality for the actor Adam, will therefore be $X_{ii}^A = 2$, where $i = 1$ (Adam). The degree centrality for the other actors and the projects can be calculated in a similar manner and these values are presented in Table A1.

*Betweenness centrality* (operationalizing junctional embeddedness) relies on the notion of geodesic paths, i.e., shortest path between two actors or events. The two-step procedure for calculating betweenness centrality involves calculating "partial betweenness" of actors first, and then using this partial betweenness to calculate actor betweenness (e.g., Freeman 1979). An actor's partial betweenness ($p_i$) is the number of pairs of actors whose geodesic paths contain the actor $i$. In case of ties, i.e., when there are multiple geodesic paths between two actors, only fractional credit is given to $p_i$, where the fraction is a reciprocal of the total number of geodesic paths between the pairs (Faust 1997). Betweenness centrality ($C_B(X_i^A)$) for this actor is then given as

$$C_B(X_i^A) = \sum_{j<k} g_{jk}(p_i)/g_{jk}, \tag{A3}$$

where $g_{jk}$ is the number of geodesic paths between actors $j$ and $k$, and $g_{jk}(p_i)$ is the number of geodesic paths between $j$ and $k$ that contain $i$.

For the information presented in Figure A1, the betweenness centrality for the actor Adam, will therefore be $\sum_{j<k} g_{jk}(p_i)/g_{jk} = 0.400$, where $i = 1$ (Adam) and $j$ and $k$ are all the other nodes, $g_{jk}(p_i)$ is the number of shortest paths between any two nodes that pass through Adam, while $g_{jk}$ is the number of possible shortest paths between any two nodes in the graph. The betweenness centrality for the other nodes can be calculated in a similar manner and these values are presented in Table A1.

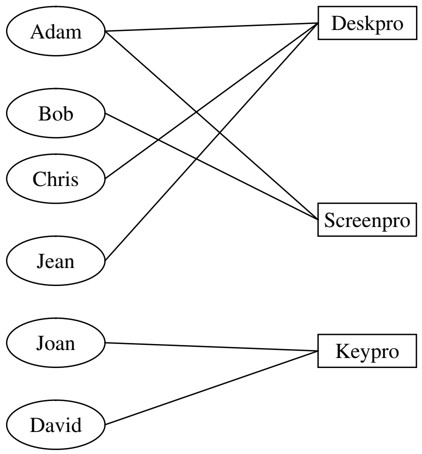
*Eigenvector centrality* (operationalizing positional embeddedness) should be high for project managers who are connected to other central project managers. Thus the eigenvector centrality for an actor depends on the strength of the ties of other actors to which this actor is connected (e.g., Faust 1997). In an affiliation network such as ours, projects can only be adjacent to developers and developers can only be adjacent to projects, which implies that the eigenvector centrality of projects is a function of the centrality of the developers associated with it, and the eigenvector centrality of developers is a function of the projects they are members in. Specifically, eigenvector centrality ($C_E(D^k)$) for a developer $D^k$ can be expressed as

$$C_E(D^k) = kC_E(P^i)\,x_{ik}, \tag{A4}$$

where $k$ is a constant, $C_E(P^i)$ is the eigenvector centrality of the project $i$ that the developer $D^k$ is a member of and strength of the tie between the developer and the project is given by $x_{ik}$. Solving for $C_E(D^k)$, which satisfies

**Figure A1** Illustrative Example of a Two-Mode Network

Consider six developers Adam, Bob, Chris, Jean, Joan, and Dave; and three projects Deskpro, Screenpro, and Keypro shown below left. No two developers share a relationship with each other directly and no two projects are linked to each other directly. However, developers share an indirect relationship through the common projects that they work on: Adam, Chris, and Jean are developers for Deskpro, and thus are related to each other. Similarly, any two projects share an indirect relationship through the developers who work on both the projects. Thus, Deskpro and Screenpro have one common developer, i.e., Adam. Adam is also the most central developer in that he has ties with three developers (with Chris and Jean because of Deskpro and with Bob because of Screenpro). Bob, a developer who works with Adam on Screenpro, is linked to Chris and Jean because of his relationship with Adam, who, in turn, shares relationships with Chris and Jean (developers of Deskpro). The affiliation graph is not fully connected, i.e., we cannot move from a project to all the other projects or from a developer to all the other developers. From this graph, one can develop the affiliation matrix shown below right, where a 1 indicates that a developer works on the project and the number 0 indicates that he does not.



**Affiliation Matrix for Network on the Left**

| Developer | Deskpro | Screenpro | Keypro |
|---|---|---|---|
| Adam | 1 | 1 | 0 |
| Bob | 0 | 1 | 0 |
| Chris | 1 | 0 | 0 |
| Jean | 1 | 0 | 0 |
| Joan | 0 | 0 | 1 |
| Dave | 0 | 0 | 1 |

the above equation for all nodes in the graph (actors and events or developers and projects) gives the eigenvector centrality for all the nodes. One can solve this system of simultaneous linear equation system by using standard the eigenvector-eigenvalue formulation. Consider a $(g+h) \times (g+h)$ sociomatrix $X$, with $g$ actors and $h$ events. Specifically, let

$$Xc = \lambda c, \qquad \text{(A5)}$$

where $\lambda$ is the largest eigenvalue, and $c$ is the vector of centrality scores. Thus the eigenvector centrality for project $P^i$ is given as (see Faust 1997)

$$C_E(P^i) = \frac{1}{\lambda} \sum_{k=1}^{h} C_E(D^k) x_{ik}, \qquad \text{(A6)}$$

where $x_{ik} = 1$ if developer $D^k$ is a member of project $P^i$ and 0 otherwise.

**Table A1** Network Centrality Measures for the Illustrative Example

| | Degree | Betweenness | Eigenvector |
|---|---|---|---|
| Project | | | |
| Deskpro | 3 | 0.467 | 0.601 |
| Screenpro | 2 | 0.267 | 0.372 |
| Keypro | 2 | 0.067 | 0.000 |
| Developer | | | |
| Adam | 2 | 0.400 | 0.512 |
| Bob | 1 | 0.000 | 0.195 |
| Chris | 1 | 0.000 | 0.316 |
| Jean | 1 | 0.000 | 0.316 |
| Joan | 1 | 0.000 | 0.000 |
| Dave | 1 | 0.000 | 0.000 |

Similarly, the eigenvector centrality of developer $D^k$ is given by the equation

$$C_E(D^k) = \frac{1}{\lambda} \sum_{i=1}^{g} C_E(P^i) x_{ik}, \qquad \text{(A7)}$$

where $x_{ik} = 1$ if developer $D^k$ is a member of project $P^i$ and 0 otherwise. Note that in case a project had more than one project manager, we added the degree, betweenness, and eigenvector centrality measures of the multiple managers to obtain the degree, betweenness, and eigenvector centrality measures for the project manager, respectively.

For the information presented in Figure A1, the eigenvector centrality for the actor Adam, will therefore be $(1/\lambda) \sum_{i=1}^{g} C_E(P^i) x_{ik} = 0.512$, where $i = 1$ (Adam). The eigenvector centrality for the other nodes can be calculated in a similar manner and these calculated values are presented in Table A1.

## References

Abrahamson, E., L. Rosenkopf. 1997. Social network effects on the extent of innovation diffusion: A computer simulation. *Organ. Sci.* **8**(3) 289–309.

Chandrashekaran, M., R. Mehta, R. Chandrashekaran, R. Grewal. 1999. Market motives, distinctive capabilities, and domestic inertia: A hybrid model of innovation generation. *J. Marketing Res.* **36**(February) 95–112.

Cooper, R. G. 2001. *Winning at New Products: Accelerating the Process from Idea to Launch*. Persues Publishing, Boulder, CO.

Crowston, K., H. Annabi, J. Howison. 2003. Defining open source software project success. *Proc. Internat. Conf. Inform. Systems, ICIS, Seattle, WA*, Association for Information Systems, Atlanta, GA.

Dayton, M. C., G. B. MacReady. 1988. Concomitant variables latent class models. *J. Amer. Statist. Assoc.* **83**(401) 173–178.

Deroian, F. 2002. Formation of social networks and diffusion of innovations. *Res. Policy* **31**(5) 835–846.

DeSarbo, W. S., W. L. Cron. 1988. A maximum likelihood methodology for clusterwise linear regression. *J. Classification* **5**(2) 249–282.

Faust, K. 1997. Centrality in affiliation networks. *Soc. Networks* **19** 157–191.

Fombrun, C. J., M. Shanley. 1990. What's in a name? Reputation building and corporate strategy. *Acad. Management J.* **33** 233–258.

Freeman, L. C. 1979. Centrality in social networks: Conceptual clarification. *Soc. Networks* **1**(3) 215–239.

Freeman, L. C., D. Roeder, R. R. Mulholland. 1980. Centrality in social networks: II. Experimental results. *Soc. Networks* **2** 119–141.

Fruchterman, T. M., E. M. Reingold. 1991. Graph drawing for force directed placement. *Software, Practice and Experience* **21** 1129–1164.

Granovetter, M. 1973. The strength of weak ties. *Amer. J. Sociol.* **78**(6) 1360–1380.

Granovetter, M. 1985. Economic action and social structure: The problem of embeddedness. *Amer. Sociol. Rev.* **15**(3) 478–499.

Granovetter, M. 1995. *Getting a Job: A Study of Contacts and Careers*, 2nd ed. University of Chicago Press, Chicago, IL.

Gulati, R. 1998. Alliances and network. *Strategic Management J.* **19**(4) 293–317.

Gulati, R., M. Gargiulo. 1999. Where do interorganizational networks come from? *Amer. J. Sociol.* **104**(5) 1439–1493.

Gupta, S., P. Chintagunta. 1994. On using demographic variables to determine segment membership in logit mixture models. *J. Marketing Res.* **16**(February) 128–136.

Lakhani, K. R., R. G. Wolf. 2003. Why hackers do what they do: Understanding motivation effort in free/open source software projects. Working Paper 4425-03, MIT Sloan School of Management, Cambridge, MA.

Laumann, E. O., P. V. Marsden, D. Prensky. 1989. The boundary specification problem in network analysis. L. C. Freeman, D. R. White, R. A. Kimball, eds. *Research Methods in Social Network Analysis*. George Mason University Press, Fairfax, VA.

Lohr, S. 2004. Speech code from I.B.M. to become open source. *New York Times* (September 13) 31–32.

Mahajan, V., E. Muller, R. A. Kerin. 1984. Introduction strategy for new products with positive and negative word-of-mouth. *Management Sci.* **30** 1389–1404.

Maltz, E., A. K. Kohli. 1996. Market intelligence dissemination across functional boundaries. *J. Marketing Res.* **33**(February) 47–61.

Mansfield, E., S. Wagner. 1975. Organizational and strategic factors associated with probabilities of success in industrial R&D. *J. Bus.* **48**(2) 179–198.

North, D. C., ed. 1990. *Institutions, Institutional Change and Economic Performance*. Cambridge University Press, New York.

O'Reilly, C. 1982. Variations in decision makers' use of information sources: The impact of quality and accessibility of information. *Acad. Management J.* **25**(4) 756–771.

Pollak, R. A., T. J. Wales. 1991. The likelihood dominance criterion. *J. Econometrics* **47** 227–242.

Portes, A. 1998. Social capital: Its origins and application in modern sociology. *Annual Rev. Sociol.* **22** 1–24.

Raghunathan, S. 1999. Impact of information quality and decision-maker quality on decision quality: A theoretical model and simulation analysis. *Decision Support Systems* **26**(4) 275–286.

Rai, A., S. S. Lang, R. B. Welker. 2002. Assessing the validity of IS success models: An empirical test and theoretical analysis. *Inform. Systems Res.* **13**(1) 50–69.

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.

Ruef, M., H. E. Aldrich, N. M. Carter. 2003. The structure of founding teams: Homophily, strong ties, and isolation among U.S. entrepreneurs. *Amer. Sociol. Rev.* **68**(2) 195–222.

Sarin, S., V. Mahajan. 2001. The effect of reward structures on the performance of cross-functional product development teams. *J. Marketing* **65**(2) 35–53.

Smelser, N. J., R. Swedberg, eds. 1994. *Handbook of Economic Sociology*. Princeton University Press, Princeton, NJ.

Spence, A. M. 1974. *Market Signaling*. Harvard University Press, Cambridge, MA.

Spencer, E. A., J. Greene. 2003. A stealth attack on Linux from Microsoft? It's paying a licensing fee to SCO—And techies smell a rat. *BusinessWeek* **31**(June 2) C4.

Titterington, D., A. Smith, U. Makov. 1985. *Statistical Analysis of Finite Mixture Distributions*. John Wiley and Sons, Chichester, UK.

Uzzi, B. 1996. The sources and consequences of embeddedness for the economic performance of organizations: The network effect. *Amer. Sociol. Rev.* **61**(August) 674–698.

Uzzi, B. 1999. Embeddedness in the making of financial capital: How social relations and networks benefit firms seeking financing. *Amer. Sociol. Rev.* **64**(August) 481–505.

Van den Bulte, C., G. L. Lilien. 2001. Medical innovation revisited: Social contagion versus marketing effort. *Amer. J. Sociol.* **106**(5) 1409–1435.

Veronesi, P. 2000. How does information quality affect stock returns? *J. Finance* **55**(2) 807–837.

von Hippel, E. 2001. Innovation by user communities: Learning from open source software. *MIT Sloan Management Rev.* **42**(4) 82–86.

von Hippel, E. 2005. *Democratizing Innovation*. MIT Press, Cambridge, MA.

von Hippel, E., G. von Krogh. 2003. Open source software and the "private-collective" innovation model: Issues for organization science. *Organ. Sci.* **14**(2) 209–225.

Wasserman, S., K. Faust. 1999. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, UK.

Wedel, M., W. A. Kamakura. 2000. *Market Segmentation: Conceptual and Methodological Foundations*. Kluwer, Boston, MA.

Williamson, O. E. 1985. *The Economic Institutions of Capitalism: Firms, Markets, and Relational Contracting*. The Free Press, New York.

Yamagishi, T., M. R. Gillmore, K. S. Cook. 1988. Network connections and the distribution of power in exchange networks. *Amer. J. Sociol.* **93**(4) 833–851.

Zukin, S., P. J. DiMaggio, eds. 1990. *Structures of Capital: The Social Organization of the Economy*. Cambridge University Press, New York.