DISCUSSION PAPER SERIES

No. 9980

NETWORK DYNAMICS AND KNOWLEDGE TRANSFER IN VIRTUAL ORGANIZATIONS: OVERCOMING THE LIABILITY OF DISPERSION

Neil Gandal and Uriel Stettner

INDUSTRIAL ORGANIZATION

Centre for Economic Policy Research

www.cepr.org

Available online at: **www.cepr.org/pubs/dps/DP9980.php**

ISSN 0265-8003

NETWORK DYNAMICS AND KNOWLEDGE TRANSFER IN VIRTUAL ORGANIZATIONS: OVERCOMING THE LIABILITY OF DISPERSION

Neil Gandal, Tel Aviv University and CEPR
Uriel Stettner, Tel Aviv University

Discussion Paper No. 9980
July 2014

Centre for Economic Policy Research
77 Bastwick Street, London EC1V 3PZ, UK
Tel: (44 20) 7183 8801, Fax: (44 20) 7183 8820
Email: cepr@cepr.org, Website: www.cepr.org

CEPR Discussion Paper No. 9980

July 2014

ABSTRACT

Network Dynamics and Knowledge Transfer in Virtual Organizations: Overcoming the Liability of Dispersion*

Product development within and across community-based and geographically dispersed virtual organizations is becoming an increasingly important mechanism through which individual knowledge holders create and disseminate knowledge in joint efforts to generate products. Without the benefits of face-to-face communication, such organizations face a particular set of constraints in their exposure to knowledge and know-how. This "liability of dispersion" increases the importance of the architecture of network ties that undergird the distinct development efforts, the embedded social structures, and the particular relationships involved in their product-generating efforts. In this paper, we examine whether particular network structures foster knowledge transfer among distinct open-source projects. We conjecture that Star developers—actors characterized by increasing levels of embeddedness and the associated ability to form ties with several projects within a network—serve boundary-spanning functions that facilitate an organization's ability to collect, assimilate, and apply external information. We find support for this conjecture in our investigation of a network of open-source software projects and developers compiled from a dataset drawn from Sourceforge.net. We also show that becoming part of a giant network component is associated with relatively large changes in project performance.

JEL Classification: L8
Keywords: exploitation, exploration, knowledge spillovers, network dynamics, open source, social capital, social network

Neil Gandal
Berglas School of Economics
Tel Aviv University
Tel Aviv 46766
ISRAEL

Email: gandal@post.tau.ac.il

For further Discussion Papers by this author see:
www.cepr.org/pubs/new-dps/dplist.asp?authorid=144974

Uriel Stettner
Recanati Business School
Tel Aviv University
Ramat Aviv, Tel Aviv 69978
ISRAEL

Email: urielste@tau.ac.il

For further Discussion Papers by this author see:
www.cepr.org/pubs/new-dps/dplist.asp?authorid=178483

*We appreciate the invaluable research assistance of Yaniv Friedensohn and gratefully acknowledge the financial support of the Israel Science Foundation (grant No. 1810112660). Any opinions expressed are those of the authors.

Submitted 15 May 2014

I. Introduction

Product development in community-based organizational settings is becoming an increasingly important mechanism through which individual knowledge holders create and disseminate knowledge in joint efforts to generate products. In its traditional form, open-source software (OSS) development is a collaborative effort of loosely coordinated and geographically dispersed developers who contribute their time and knowledge to establishing and improving software and whose underlying knowledge is made accessible to the general population.

OSS generally implies that a particular computer software source code is available to the broad public under an OSS license (Laurent, 2004.). Such licenses grant the rights to use an entire work, to create a derivative work, or to share or market such work subject to the license governing the specific open-source project (Bonaccorsi, Rossi, & Giannangeli, 2006; Von Hippel & Von Krogh, 2003; Lerner & Tirole, 2002). Accordingly, one of the central aspects of OSS development is the project's ability to share and absorb knowledge that has been created within or outside of a distinct OSS project. Such spillovers facilitate the transfer of knowledge and ideas within and across researchers and development teams. External knowledge may provide a particular project with highly specialized competencies and technical flexibility through the formation of informal "learning alliances" that may result in accelerated learning processes and a contraction of the product development life cycle, which means stronger value proposition.

Clearly, the evolving social structure that underlies distinct OSS development efforts is a critical point of distinction from traditional proprietary, closed-innovation development mechanisms. The open-source structure emphasizes the significance of social capital in defining organizational traits such as the accessibility of diverse knowledge, the aptitude to recruit qualified human capital (Lacetera, Cockburn, & Henderson, 2004), and/or the capacity to increase product visibility and increase adoption rates (Burt, 1992; Granovetter, 1985, 2005; Uzzi & Gillespie, 2002). This architecture of network ties offers a glimpse into the extent to which an entity is a) rooted in a network, b) connects with other entities, and c) connects with other structurally embedded entities. Accordingly, an entity that is characterized by higher levels of embeddedness is expected to possess higher levels of social capital, which should, in turn, exert a positive impact on both the technical and commercial successes of the open-source project with which the entity is associated (Grewal, Lilien, & Mallapragada, 2006).

We contribute to research on organizational learning by studying how changes in network structures can foster knowledge transfers. We show that changes in the network architecture are associated with changes in project success. We also find that actors who are characterized by high levels of embeddedness serve boundary-spanning functions that facilitate a project's ability to collect, assimilate, and apply external information. Thus, we demonstrate the positive performance implications of collaboration across project boundaries.

As network structures evolve over time, projects connect (and disconnect) from one another. Typically, mature network structures reflect one giant and many small components Thus, we also can compare organizations that were in the giant component of a network for relatively long periods of time with organizations that joined such giant components during the sample period. We find that established projects within the giant component benefit differently from changes in network structures than projects that only recently entered such giant component of a network.

Some recent studies have examined the relationship between network structure and behavior (e.g., Ballester, Calvó-Armengol, & Zenou, 2006; Calvo-Armengol & Jackson, 2004; Jackson & Yariv, 2007; Karlan, Mobius, Rosenblat, & Szeidl, 2009) or performance (Ahuja, 2000; Calvó-Armengol, Patacchini, & Zenou, 2009). Our paper is closest to that of Fershtman and Gandal (2011), who focus on spillovers that occur by means of the interactions of different researchers or developers in OSS projects. Using cross-sectional data, these authors demonstrate that the structure of the product network is associated with the project's success, which provides support for knowledge spillovers. Nevertheless, no paper discussed above focuses on the relationship between changes in the network architecture and changes in success over time, which is the focus of our paper.[2]

In the following section, we provide the theoretical foundations for our empirical work. Based on these foundations, we develop several hypotheses that we test empirically.

II. Theoretical Foundations and Hypotheses

Virtual teams are semi-structured groups of geographically dispersed and skilled individuals working on interdependent tasks using informal, non-hierarchical, and decentralized communication with the common goal of creating a valuable product (Lipnack & Stamps, 1997). Virtual teams, as opposed to traditional work teams that enjoy the benefits of face-to-face communication, encounter a particular set of challenges that negatively impact a team's ability to form personal relationships (Beyerlein, Johnson, & Beyerlein, 2001), team communication (Pinto & Pinto, 1990), and performance and satisfaction (Jehn & Shah, 1997). The resulting lack of strong connections (Wong & Burton, 2000) impacts commitment (Whiting & Reardon, 1998), whereas the lack of social support has a negative effect on productivity (Cascio, 2000; Townsend, DeMarie, & Hendrickson, 1998) through reduced willingness to share knowledge,

[2] Goyal, van der Leij and Moraga-Gonzalez (2006) constructed a co-authorship network using data on published papers that were included in EconLit between 1970 and 2000 to study network properties over time. Nonetheless, they were interested in different issues than we are.

trust, creative output, and leadership. Accordingly, by the nature of its organizational design and structure, members of dispersed development teams are restricted in their exposure to knowledge and know-how. However, these organizations can benefit from cooperation among actors in the social network and can enhance development teams' cohesiveness and growth (Borgatti, Jones, & Everett, 1998; Chung, Singh, & Lee, 2000). Indeed, an entity that is characterized by higher levels of embeddedness should possess higher levels of social capital and, in turn, demonstrate a positive impact on both the technical and commercial successes of the open-source project with which the entity is associated (Grewal et al., 2006).

II. Hypotheses

The performance implications of knowledge Spillovers via Star Contributors

If social capital is the lock with respect to knowledge networks, who holds the key with which to disperse knowledge and facilitate knowledge spillovers? We argue that a development team's ability to innovate depends on developers with distinct network characteristics who have boundary-spanning abilities (Rothaermel & Hess, 2007). In particular, developers who work in multiple organizations can take software code from the other organizations in which they work. We define developers who work on a large numbers of projects as 'Stars.' Because they are boundary spanners, Stars can bridge organizational and environmental boundaries to identify novel knowledge and evaluate, streamline, and organize knowledge flows from external sources (Cohen & Levinthal, 1990; De Jong & Freel, 2010). A developer who works in a large number of organizations has a greater capacity to access external knowledge. Stars thus facilitate an organization's ability to collect, assimilate, and apply external information.

In community-based organizational settings, Stars are contributors who work on multiple projects and thus play important roles in knowledge spillovers. Clearly, having a Star contributor join a project increases the potential for access to external knowledge and positive spillovers across a project. Moreover, the broader that the potential reach (i.e., scope) of Star developers is to external organizations and developers, the greater the potential is for novel and valuable external knowledge access and positive spillovers. As Stars mature with respect to their network positions over time, the scope of their reach across distinct organizations is enhanced such that they enjoy greater potential access to external knowledge and positive spillovers. Thus, Star developers can improve the innovative output of development teams and, in turn, exert a positive impact on both the technical and commercial successes of the product. In summary, Star developers are actors characterized by higher levels of embeddedness and the associated ability to transfer knowledge across organizations within a network. Accordingly, it is expected that Stars have higher social capital and occupy boundary-spanning functions that facilitate their ability to collect, assimilate, and apply external information.

Hypothesis 1: Star developers, through their exposure to external organizations and developers, have a positive impact on an organization's performance beyond the associated network structures they create.

The performance implications of knowledge Spillovers via Contributors

The characteristics of knowledge and its exchange properties are best captured in the distinction between tacit and explicit knowledge (Nonaka, 1994). Tacit knowledge is not easily shared, communicated, or codified and is acquired through experience and skill (Polanyi, 1967). Tacit knowledge is highly personal, is deeply rooted in both action and an individual's commitment to a specific context, and consists of technical skills, mental models, beliefs, and perspectives (Nonaka et al., 1995). Explicit knowledge, however, can be more easily expressed, captured, stored, and reused through text or speech. Tacit knowledge is important for innovation, and converting tacit knowledge into explicit knowledge is essential to creating organizational knowledge. Tacit knowledge can be transferred among individual knowledge holders through observation, imitation and practice, at which point it can be reconfigured and diffused to the development team at large (Nonaka, 1995). How can teams gain access to tacit knowledge across distinct organizations? Organizational members—both Stars and non-Stars—can join other organizations and spread knowledge from one project to another. With OSS, this result typically ensues in the form of bringing code (or variations of code) from one project to another. The shorter the distance that a particular project is from other projects, the greater the amount of knowledge that will spread to such project.

> **Hypothesis 2:** Organizations with high closeness centrality enjoy greater ease with which developers can absorb and diffuse knowledge within and across organizations and, thus enhance performance.

Two organizations are directly connected if they have at least one contributor in common. In such a case, the knowledge transferred directly by a particular contributor might be more tacit and of greater value than explicit knowledge that is transferred indirectly via several contributors. Thus, directly connected projects potentially receive more tacit knowledge from their neighbors. By engaging in boundary-spanning activities, organizations can internalize and leverage resources and capabilities that are distant from their core competencies (Rosenkopf & Almeida, 2003). Searching beyond organizational boundaries enables the discovery of opportunities that are unavailable internally and that are situated beyond the span of local search. In fact, such variety-seeking supports project flexibility and innovation, which are characteristics that are associated with enhanced performance (Burgelman, 2002; March, 1991).

Hypothesis 3: Organizations with high degree centrality enjoy greater direct access to tacit knowledge that enhances organizational performance.

Effect of Joining the Giant Component

Which network structures are most conducive to providing boundary-spanning activities? Network structures evolve over time as organizations connect (and disconnect) from one another. Typically, mature network structures reflect one giant and many small components. In turn, the dynamics of network formation will generate the following two different sets of organizations in a giant component: (i) organizations that were in the giant component throughout a specified period of time and (ii) organizations that joined the giant component sometime during such specified period of time. What differences are there between these two sets of organizations? Members of the giant component have greater access to knowledge beyond project boundaries. Thus, we expect that projects that move into the giant component would receive a substantial benefit from being exposed to spillovers within the giant component.

Hypothesis 4: The addition of a Star will do more for a project that moves into the giant component of a network than for an organization already in such giant component.

Hypothesis 5a: Associations between the change in closeness and the change in an organization's performance will be stronger for organizations moving into the giant component compared with organizations that have always been in the giant component.

Hypothesis 5b: Associations between the change in degree and the change in an organization's performance will be stronger for organizations moving into the giant component compared with organizations that have always been in the giant component.

III. Methods

Research Setting and Data

This paper uses a replica of publicly available data from Sourceforge.net that is hosted at Notre Dame University. Sourceforge.net facilitates software developer collaboration by providing a free online platform for managing projects, communications, and software code. Sourceforge.net is the largest repository of registered OSS development projects during the period of our study.

Each SourceForge.net project contains a list of registered team members who contribute their time and knowledge to the advancement of an OSS project. Each project links to a "developer page" that contains meta information on a particular contributor, including the date the developer joined the project, the developer's functional description (e.g., administrator, developer) and his or her geographic location. These projects are managed by project administrators. Because accessibility to the OSS projects is unrestricted and because the contributors can be identified by their unique user names, we utilize this information to construct a two-mode network that relates projects via registered contributors. Accordingly, we define two OSS projects as being connected when there are common contributors who participate in both projects.[3]

[3] We assume that project members are added to the list because they make a contribution to the project that involves an investment of time and effort. A project is thus understood as a collaborative effort by its contributors.

Each project links to a standardized “Project page” that lists descriptive information on a particular project, including a statement of purpose, the intended audience, the license, and the operating system for which the application is designed. Moreover, a “Statistics page” shows various project activity measures, including the number of project page views and downloads registered for the focal project. Although some data are available for other periods, statistics on downloads are available only for the 2006–2009 period.[4] Therefore, we deploy panel data from 2006–2009 to construct two distinct two-mode networks: (i) the project network and (ii) the contributor network. In the former, the nodes are the OSS projects, and two projects are linked when there are common contributors who work on both. In the latter, the nodes of the contributor network are the contributors, and two contributors are linked if they participated in at least one OSS project together.

Regarding the project network in 2009, we find that 84.3% percent of the projects have either one or two contributors, 9.2% have three to four contributors and 6.5% have five or more contributors (see Table 1). With regard to the contributor network in January 2009, 91.3% of the contributors worked on one or two projects, 6.5% of the contributors worked on three to four projects, and 2.1% of the contributors worked on five or more projects.[5] In our analysis, we focus on the project network, but we also include a key feature of the contributor network in the analysis: contributors who work on five or more projects. We define such contributors as ‘Stars.’ As we will see, these ‘Stars’ play an important role in knowledge spillovers.

[4] Page view data are not available over time, but page views are highly correlated with downloads.
[5] These percentages were virtually identical in 2006 as well.

Table 1: Distribution of components in project networks—January 2009

Project Network	Contributor Network		
Contributor s per project	*Percent of total projects*	*Projects per contributor*	*Percent of Contributors*
1	69.9	1	77.2
2	14.4	2	14.1
3-4	9.2	3-4	6.5
5-9	4.8	5-9	1.9
10 or more	1.7	10 or more	0.2

Variables

Having panel data from 2006 to 2009 allows us to focus on differences over time. This is helpful because it is difficult to determine causality from cross-sectional data, and, therefore, unobserved fixed project effects might be driving success. Because we do not have data on these fixed project effects, they are included in the error term when running cross-sectional analyses. If these unobserved effects are correlated with the right-hand-side variables, the estimates from the cross-sectional analysis will be biased; however, this problem is eliminated when using data on differences over time.

Dependent Variable

We wish to examine whether knowledge spillovers play a significant role in the development of OSS projects and evaluate the importance of Stars. Consistent with prior research, we measure project performance by examining the number of times a project has been downloaded (Fershtman & Gandal, 2011; Grewal et al., 2006). We focus on downloads of the executable, compiled product because users will not typically download the source code. We define ΔDownloads as the difference between the total number of downloads in January 2009 and January 2006. We further define lΔDownloads ≡ ln(1+ΔDownloads), where "ln" means the natural logarithm, and Δ is the difference operator.

Independent Variables

Knowledge spillovers from project to project occur via individuals. In the case of OSS projects, contributors frequently port code that is embedded in one project into another project to which they contribute. Direct spillovers occur when projects have a common developer who transfers information and knowledge (primarily code) from one project to another. Project spillovers may also be indirect, i.e., when knowledge is transferred from one project to another when the two projects are not directly linked (there is no common contributor). Because we do not directly observe spillovers, we will examine the relationship between the network structure and project success to identify the relative importance of knowledge spillovers.

We define two network centrality measures: (i) a project's *degree* is defined as the number of projects with which the focal project has a direct link or common developers and (ii) a project's *closeness centrality*, which is defined as the inverse of the sum of all distances between a focal project and all other projects multiplied by the number of other projects.[6] Intuitively, closeness centrality measures how far each project is from all the other projects in a network.[7]

Accordingly, we define ***ΔDegree*** as the difference in the degree centrality the project between January 2009 and January 2006. Similarly, we define **ΔClose** as the difference in the closeness centrality of the project between January 2009 and January 2006.

---

[6] See Freeman (1979), pp. 225-226 and Faust and Wasserman (1994), pp. 184-185 for details on how closeness centrality is calculated.

[7] Closeness centrality lies in the range [0,1]. In the case of a Star network with a single project in the middle that is connected to all other projects, the closeness centrality of the project in the center is one.

Next, we define **ΔC*pp*** as the change in the number of contributors that participated in the project during the three-year period from January 2006 to January 2009. Since the number of contributors might fall or rise over time, ΔDegree, ΔClose, and ΔCpp can be either positive or negative.

In addition to downloads and the project network variables described above, we have data for a group of control variables. In Sourceforge.net, projects evolve through six stages, beginning with planning (1) and continuing to pre-alpha (2), alpha (3), beta testing (4), production (5), and finally maturity (6). We define a dummy variable, Δstage, that assumes the value one if there was stage progression (e.g., from alpha (3) to production (4)) and zero if there was no change in stage.

To control for the amount of time that the project has been in existence, we define the variable years_since as the number of years that have elapsed since the project first appeared at Sourceforge.net: lyears_since = ln(years_since).

Finally, we define a Star as a contributor who worked on five or more projects. This variable comes from the contributor network, not the project network. Clearly, having a "Star" contributor join a project gives that project more connections to other projects. An interesting question is whether adding a "Star" to the team of developers has an effect on the success of a project. To examine this effect, we include a variable, denoted as ΔStar5, which can take on positive or negative values and is defined as the change in the number of Stars on a project from 2006 to 2009.

Discussion of the Data

In our panel data set, we have 42,796 projects with complete information.[8] Complete information indicates that the projects existed in both 2006 and 2009 and that we have data for all the relevant variables discussed above. We exclude observations for Δdegree, Δcloseness, and ΔCpp that are (approximately) in the lowest 5% of these distributions. Specifically, we exclude 961 observations of Δdegree that are greater than or equal to -4, an additional 394 observations of ΔCpp that are greater than or equal to -1, and an additional 557 observations of Δcloseness that are less than -0.0037. We exclude these observations because large negative changes in Cpp, degree, and closeness might simply be explained by those particular projects being more likely to remove any inactive programmers from their projects' websites in comparison with other projects. Our results are also robust to including all 42,796 observations. We report these results in the appendix.

After excluding the 1,912 projects discussed above, we are left with 40,884 observations for the analysis. Approximately one-third of the projects in the main part of the paper (13,474) are in the giant component, and the second-largest component is small (64 projects.). This distribution (one giant component and many small components) is typical of many networks.

Particularly interesting are the 2,656 projects that were not in the giant component in 2006 but were included in the giant component in 2009. These projects comprise 20% of the giant component. Not surprisingly, these observations exhibit relatively large changes in degree, Cpp, closeness, stage and Stars. An interesting question is whether these projects have different properties than other projects in the giant component.

[8] Importantly, because we have data on the participants in every project, our networks are constructed using all projects, including projects without complete information.

Descriptive statistics are shown in Table A1 in the appendix. The mean and median download changes for projects in the giant component (mean = 66,819 and median = 930) is much greater for projects in the giant component than for projects outside of the giant component (mean = 20,734 and median = 373).

When we compare the two subgroups within the giant component—namely the projects in the giant component throughout the 2006–2009 period and the projects that moved into the giant component during the 2006–2009 period—we find no difference in the mean or median changes in downloads among the groups. Projects that moved into the giant component have much higher changes in degree, closeness, and the number of Stars than projects in the giant component throughout the 2006–2009 period (see Table A1).

Correlations between changes in degree, closeness, Stars, and Cpp are all relatively low, as shown in Table A2 of the appendix. The highest correlation is between ΔCpp and Δdegree, but that correlation is only 0.53. No other correlation exceeds a magnitude of 0.34.

VI. Empirical Analysis:

The relationship between the number of contributors and *downloads* is likely non-linear: additional contributors are likely associated with a larger number of downloads, but the marginal effect of each additional contributor declines as the number of contributors increases. The same is likely true for the relationship between network variables and downloads as well, which suggests that a "log/log" model is appropriate.[9] Thus, we use the following estimating equation:

[9] We estimate a log/log specification. As with the case of downloads, all independent variables (except changes in the number of Stars and changes in stage) are in logarithmic form, and we denote this situation by including an 'l' before the variable name—e.g., lΔCpp is the logarithm of the change in the number of contributors. We add a constant to lΔCloseness, lΔCpp, and lΔDegree such that the logarithm is defined.

$$[1] \quad l(\Delta \text{Downloads}) = \beta_0 + \beta_1 (l\Delta \text{Cpp}) + \beta_2 (l\Delta \text{Degree}) + \beta_3 (l\Delta \text{Close}) + \beta_4 (\Delta \text{Star5}) + \beta_5 (\Delta \text{Stage}) + \beta_6 (\text{lyears\_since}) + \varepsilon,$$

where Δ is the difference operator and ε is a white-noise error term. [10] We estimate [1] for the following four cases:

Case I:	Projects outside of the giant component
Case II:	Projects in the giant component in January 2009
Case IIA:	Projects in the giant component throughout the 2006–2009 period
Case IIB:	Projects that moved into giant component during the 2006–2009 period

Knowledge spillovers via Star contributors

Hypothesis 1 proposed that Star developers have a positive impact on product success. Table 2 shows that a change in the number of Stars does not significantly influence downloads for projects outside the giant component (Case I: $\beta = -0.016$, $p = 0.81$). However, changes in the number of Stars is significantly positively associated with changes in the number of downloads for projects that are in the giant component (Case II: $\beta = 0.14$, $p = 0.01$). Thus, in support of Hypothesis 1, changes in the number of Stars is positively associated with changes in the number of downloads in the giant component even after controlling for the network structure. This effect does not exist for projects outside the giant component, which suggests that the spillovers via Stars are due in part to being in the giant component.

[10] We examine alternative functional forms as well. Not surprisingly, we find that the log/log specification has a much higher adjusted R-squared than the log/linear specification and a linear/linear specification performs even more poorly.

To test Hypothesis 4, we compare the impact of Star developers who were in the giant component consisting of 10,818 projects throughout the entire period of the study (Case IIA) to those associated with the 2,656 projects who later joined the giant component sometime between January 2006 and 2009 (Case IIB). Table 2 shows that, whereas changes in the number of Stars on a project is not significantly associated with changes in downloads for projects that moved into the giant component (Case IIB: $\beta = 0.064$ $p = 0.60$), changes in the number of Stars is significantly positively associated with changes in the number of downloads for projects that were always in the giant component (Case IIA: $\beta = 0.14$, $p = 0.04$). Although this result initially surprised us, we believe that the result can be explained as follows: the more prolonged that the exposure of projects to external projects and developers is, the greater the positive impact that the addition of a Star has on project success.

Knowledge Spillovers via contributors

Case II in Table 2 also shows that changes in closeness centrality are positively associated with changes in project performance, which supports Hypothesis 2.[11] Table 2 also shows that changes in closeness are positively and significantly associated with changes in downloads for both the projects that moved into the giant component (Case IIB: $\beta = 0.89$, $p < 0.01$) and the projects that were always in the giant component (Case IIA: $\beta = 0.15$, $p < 0.0001$). However, the effect is much stronger for the projects that moved into the giant component, which supports Hypothesis 5a. Spillovers on project success are particularly pronounced for late bloomers, which suggests that joining a large pool of knowledge (i.e., the giant component) allows projects to gain access to high-impact, novel knowledge and ideas.

[11] Recall that when we employ closeness in the analysis, we must restrict attention to connected projects.

Hypothesis 3 argues that a high degree centrality is associated with enhanced project success. Table 2 shows that a change in the degree centrality is indeed positively associated with a change in the number of downloads for projects outside the giant component (Case I: $\beta = 0.42$, $p<0.0001$) and projects in the giant component (Case II: $\beta = 0.34$, $p < 0.0001$), which provides support for Hypothesis 3.

In fact, table 2 also shows that the effect is approximately twice as large for the projects that moved into the giant component (Case IIB: $\beta = 0.63$, $p < 0.0001$) than for the projects that were always in the giant component (Case IIA: $\beta = 0.30$, $p < 0.0001$). Thus, we find support for Hypothesis 5b. These projects enjoy enhanced boundary-spanning capacities, thereby enabling the organization to discover opportunities that are unavailable internally and that exist outside of the reach of a local search.

Table 2 also shows that changes in the number of contributors are positively associated with changes in the number of downloads. This association is true for projects outside the giant component (Case I) and projects in the giant component (Case II). When we split the giant component into two groups, we see that this result holds as well for projects always in the giant component (Case IIA) and projects that moved into the giant component between 2006 and 2009 (Case IIB).

Table 2: Main results

DV: lΔdownloads	**Case I** Outside the Giant Component	**Case II** In the Giant Component	**Case IIA** Always in the Giant Component	**Case IIB** Moved into the Giant Component
Constant	6.30 (32.42)	5.00 (14.95)	4.71 (12.92)	7.65 (5.53)
lΔCpp	1.32 (18.83)	1.73 (33.33)	1.80 (30.19)	1.42 (13.25)
lΔdegree	0.42 (7.07)	0.34 (6.53)	0.30 (5.39)	0.63 (3.76)
lΔcloseness		0.15 (4.41)	0.15 (4.22)	0.89 (2.73)
ΔStars5	-0.016 (-0.25)	0.14 (2.48)	0.14 (2.08)	0.064 (0.53)
Δstage	1.04 (17.07)	0.92 (10.56)	0.99 (8.97)	0.76 (5.53)
lyears_since	-0.87 (-12.41)	0.41 (3.69)	0.54 (4.18)	-0.04 (-0.18)
Moved into Giant Component		-0.59 (-6.41)		
# of Observations	27,410	13,474	10,818	2,656
Adjusted R-squared	0.04	0.14	0.14	0.16

VII. Robustness Tests

Projects with more than one contributor

We repeat the analysis for projects with more than one contributor. Table 3 shows that all of the main results discussed above continue to hold; thus, our results are robust to excluding projects with just a single contributor. The result for Stars has borderline significance in Case IIA; again, however, Stars seem to matter more for projects that have benefitted from being in the giant component for a relatively long period of time than for projects that moved into the giant component more recently (Case IIB).

Table 3: Projects with more than one contributor

Dept Variable: lΔdownloads	**Case I** Outside the Giant Component	**Case II** In the Giant Component	**Case IIA** Always in the Giant Component	**Case IIB** Moved to the Giant Component
Constant	6.00 (16.46)	4.67 (10.78)	4.35 (9.24)	8.25 (4.51)
lΔCpp	1.51 (17.54)	1.54 (26.88)	1.61 (24.94)	1.18 (9.31)
lΔdegree	0.41 (4.28)	0.37 (6.25)	0.33 (5.31)	0.65 (3.03)
lΔcloseness		0.15 (3.42)	0.14 (3.24)	1.08 (2.45)
ΔStars5	-0.036 (-0.32)	0.14 (1.89)	0.13 (1.62)	0.062 (0.39)
Δstage	0.89 (8.50)	0.75 (7.11)	0.80 (6.15)	0.63 (3.61)
lyears_since	-0.61 (-4.38)	0.77 (5.21)	0.90 (5.37)	0.23 (0.74)
Moved into Giant Component		-.060 (-5.10)		
# of Observations	8,094	8,632	7,061	1,571
Adjusted R-squared	0.07	0.15	0.15	0.15

In Table A3 in the appendix, we include all observations. Although the R-squared coefficients are much smaller in the regressions in Table A3 than in Table 2, the results are qualitatively unchanged, which greatly strengthens the main results of the paper.

VIII. Testing For Endogeneity

Although our discussion focuses on how the network structure affects success, the reverse may be true as well: contributors may want to join popular projects. Developers may want to be associated with more successful projects, thereby making the number of contributors (and thus the degree) endogenous.[12] In fact, the Sourceforge.net website states that, “as a project's activity rises, SourceForge.net's internal ranking system makes it more visible to other developers who may join and contribute to it. Given that many open-source projects fail due to a lack of developer support, exposure to such a large community of developers can continually breathe new life into a project.”

Here, we discuss the tests that we employ to investigate potential endogeneity. Because we are using panel data with network variables, several approaches to test for the endogeneity of *Cpp, degree* and *closeness* are possible. We believe that the most convincing test for endogeneity is to restrict ourselves to those projects that had no changes in the number of contributors over the 2006–2009 period. In such a case, reverse causality (i.e., the effect that describes the tendency to join popular projects) is absent.[13] Note that the degree can change for projects that have no changes in the number of their contributors. The mechanism by which this change can

[12] Closeness can also be endogenous, but only under an unlikely scenario. Nevertheless, we test for endogeneity here as well.

[13] Of course, it is possible that some contributors joined and some left with a net change of zero, but the overwhelming majority of projects had no changes in personnel.

occur is that the degree centrality of the original project also increases when a contributor on a particular project joins another project.[14]

Our results describing what occurs when we restrict the analysis to projects that had no change in the number of contributors are reported in Table 4 for Cases II, IIA, and IIB. As expected, we find that the effect of changes in closeness on changes in downloads is completely robust to all these ‘tests’ for endogeneity, which is not surprising because closeness can only be endogenous under an unlikely scenario. Similarly, the results regarding Stars are virtually unchanged from the results provided in Table 2.

In the case of degree, a comparison between Tables 2 and 4 shows that the results for degree are slightly smaller in Table 4 because of the 'joining popular projects effect.' Nevertheless, in all three cases (II, IIA, and IIB,) the estimated coefficients for degree are statistically significant. This analysis suggests that reverse causality is not driving the results.

Table 4: Testing for Endogeneity

Dept Variable: lΔdownloads	Case II: In the Giant Component: ΔCpp = 0	Case IIA: Always in the Giant Component ΔCpp = 0	Case IIB: Moved into the Giant Component ΔCpp = 0
Constant	7.28 (19.41)	7.13 (17.48)	9.06 (6.09)
lΔCpp			
lΔdegree	0.25 (3.98)	0.22 (3.35)	0.50 (2.53)
lΔcloseness	0.17 (4.54)	0.17 (4.39)	0.81 (2.21)
ΔStars5	0.17 (2.53)	0.16 (2.10)	0.11 (0.82)
Δstage	0.86 (8.71)	1.31 (8.80)	0.93 (4.81)
lyears_since	-0.077 (-0.62)	0.0043 (0.03)	-0.38 (-1.47)
Moved to Giant Component	-0.64 (-6.30)		
# of Observations	10,421	8,578	1,843
Adjusted R-squared	0.02	0.02	0.03

[14] Similar to degree, the number of Stars on a project can change even when the number of contributors does not, which occurs when a contributor on one project joins other projects and transitions from working on fewer than five projects to working on five or more projects.

IX. Discussion:

Prior research studying the relationship between network structure and performance has ignored the implications of the dynamics of knowledge spillovers that occur by means of the interaction of different developers collaborating in different research projects over time. We contribute to the research on organizational learning by studying how network structures can foster knowledge transfer that occurs through developers interacting across distinct development projects. We show that changes in the network architecture are indeed associated with changes in project success.

Our findings reveal that actors characterized by increasing levels of embeddedness and the associated ability to transfer knowledge across projects within a network have higher levels of social capital and serve boundary-spanning functions that facilitate a project's capacity to collect, assimilate, and apply external information. We demonstrate the positive performance implications of collaboration across project boundaries and link the benefits to knowledge spillovers across projects. In fact, we show that the shorter the distance is between projects, the greater will be the ease with which developers can absorb and diffuse knowledge within and across projects and thus enhance performance.

We further demonstrate the benefits that the members—in particular, the Stars—of giant components enjoy when seeking to leverage their access to knowledge beyond project boundaries. Thus, network structures are conducive to providing boundary-spanning activities. In fact, the dynamics of network formation reflect additional benefits to organizations that were in the giant component throughout an extended period of time, which suggests that projects that move into the giant component enjoy the substantial benefit of being exposed to spillovers within the giant component. Thus, the more prolonged the exposure of developers to external projects is, the greater will be the positive impact of such exposure on project performance.

Our study advances the understanding of the link between network structures, agent network position, and organizational performance; nevertheless, it is subject to a few limitations. First, we have theorized about Stars' capacities to access, assimilate, and diffuse explicit and tacit knowledge via boundary-spanning activities. Future research should attempt to measure these latent variables that underlie the innovativeness and productivity of development teams.

Second, we have not fully studied the characteristics and capabilities of Star developers that allow these individuals to develop into Stars and that facilitate exploitation of their idiosyncratic resources. Future research may take a deeper look at the importance of Star developers. Given the strategic importance to project success of having Star developers, such research might seek to identify distinct demographic characteristics of Stars and their relative contributions to projects. Thus, such research could evaluate the technological overlap between the skill sets of Stars and their peers within distinct projects and then mighty study the relative importance of connected projects and the directionality of knowledge flow.

Finally, our sample is limited to the software industry; however, the concept of crowdsourcing has been applied to other industries, including hardware, biology, and astronomy, to name a few. Future research may generalize our findings to other industries. The software industry relies extensively on the internalization of external knowledge to complement internally developed products. Future research may study the link between network structures, the agents within such structures, and performance in industries in which product development relies on specialized knowledge.

In spite of these limitations, we believe that we have helped advance research on organizational learning by relating changes in the network architecture to changes in project success and by revealing that actors characterized by increasing levels of embeddedness possess boundary-spanning functions that facilitate a project's capacity to collect, assimilate, and apply external information. In so doing, we demonstrate the positive performance implications of collaboration across project boundaries.

References

Ahuja, G. 2000. Collaboration Networks, Structural Holes, and Innovation: A Longitudinal Study. ***Administrative Science Quarterly***, 45(3): 425–455.

Ballester, C., Calvó-Armengol, A., & Zenou, Y. 2006. Who's who in networks. wanted: the key player. ***Econometrica***, 74(5): 1403–1417.

Beyerlein, M. M., Johnson, D. A., & Beyerlein, S. T. 2001. ***Virtual teams***. Jai.

Bonaccorsi, A., Rossi, C., & Giannangeli, S. 2006. Adaptive entry strategies under dominant standards: Hybrid business models in the Open Source software industry. ***Management Science***, 52(7): 1085–1098.

Borgatti, S. P., Jones, C., & Everett, M. G. 1998. Network measures of social capital. ***Connections***, 21(2): 27–36.

Burgelman, R. A. 2002. Strategy as vector and the inertia of coevolutionary Lock-in. ***Administrative Science Quarterly***, 47(2): 325–357.

Burt, R. S. 1992. ***Structural holes: the social structure of competition***. Harvard University Press.

Calvo-Armengol, A., & Jackson, M. O. 2004. The effects of social networks on employment and inequality. ***The American Economic Review***, 94(3): 426–454.

Calvó-Armengol, A., Patacchini, E., & Zenou, Y. 2009. Peer effects and social networks in education. ***The Review of Economic Studies***, 76(4): 1239–1267.

Cascio, W. F. 2000. Managing a virtual workplace. ***The Academy of Management Executive***, 14(3): 81–90.

Chung, S., Singh, H., & Lee, K. 2000. Complementarity, status similarity and social capital as drivers of alliance formation. ***Strategic Management Journal***, 1–22.

Cohen, W. M., & Levinthal, D. A. 1990. Absorptive capacity: A new perspective on learning and innovation. ***Administrative Science Quarterly***, 35(1): 128–152.

Fershtman, C., & Gandal, N. 2011. Direct and indirect knowledge spillovers: the "social network" of open-source projects. ***The RAND Journal of Economics***, 42(1): 70–91.

Goyal, S., Van Der Leij, M. J., & Moraga-González, J. L. 2006. Economics: An emerging small world. ***Journal of Political Economy***, 114(2): 403–412.

Granovetter, M. 1985. Economic action and social structure: the problem of embeddedness. ***American Journal of Sociology***, 91(3): 481–510.

Granovetter, M. 2005. The impact of social structure on economic outcomes. ***Journal of Economic Perspectives***, 19(1): 33–50.

Grewal, R., Lilien, G. L., & Mallapragada, G. 2006. Location, location, location: How network embeddedness affects project success in open source systems. ***Management Science***, 52(7): 1043–1056.

Hannan, M. T., & Freeman, J. 1984. Structural inertia and organizational change. ***American Sociological Review***, 49(2): 149–164.

Von Hippel, E., & Von Krogh, G. 2003. Open source software and the" private-collective" innovation model: Issues for organization science. ***Organization Science***, 14(2): 209–223.

Jackson, M. O., & Yariv, L. 2007. Diffusion of behavior and equilibrium properties in network games. ***The American economic review***, 97(2): 92–98.

Jehn, K. A., & Shah, P. P. 1997. Interpersonal relationships and task performance: An examination of mediation processes in friendship and acquaintance groups. ***Journal of Personality and Social Psychology***, 72(4): 775.

De Jong, J. P. J., & Freel, M. 2010. Absorptive capacity and the reach of collaboration in high technology small firms. ***Research Policy***, 39(1): 47–54.
Karlan, D., Mobius, M., Rosenblat, T., & Szeidl, A. 2009. Trust and social collateral. ***The Quarterly Journal of Economics***, 124(3): 1307–1361.
Lacetera, N., Cockburn, I. M., & Henderson, R. 2004. Do firms change capabilities by hiring new people? A study of the adoption of science-based drug discovery. ***Advances in strategic management***, 21: 133–159.
Laurent, A. M. . 2004. ***Understanding open source and free software licensing***. O'Reilly Media, Inc.
Lerner, J., & Tirole, J. 2002. Some simple economics of open source. ***The journal of industrial economics***, 50(2): 197–234.
Lipnack, J., & Stamps, J. 1997. ***Virtual teams: Reaching across space, time, and organizations with technology***. John Wiley & Sons Inc.
March, J. G. 1991. Exploration and exploitation in organizational learning. ***Organization Science***, 2(1): 71–87.
Nonaka, I. 1994. A dynamic theory of organizational knowledge creation. ***Organization Science***, 5(1): 14–37.
Nonaka, I. 1995. ***The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation: How Japanese Companies Create the Dynamics of Innovation***. Oxford university press.
Pinto, M. B., & Pinto, J. K. 1990. Project team communication and cross-functional cooperation in new program development. ***Journal of Product Innovation Management***, 7(3): 200–212.
Polanyi, M. 1967. The tacit dimension. ***Knowledge in organizations***, 135–146.
Rosenkopf, L., & Almeida, P. 2003. Overcoming local search through alliances and mobility. ***Management Science***, 49(6): 751–766.
Rosenkopf, L., & Nerkar, A. 2001. Beyond local search: Boundary-spanning, exploration, and impact in the optical disk industry. ***Strategic Management Journal***, 22(4): 287–306.
Rothaermel, F. T., & Hess, A. M. 2007. Building Dynamic Capabilities: Innovation Driven by Individual-, Firm-, and Network-Level Effects. ***Organization Science***, 18(6): 898–921.
Townsend, A. M., DeMarie, S. M., & Hendrickson, A. R. 1998. Virtual teams: Technology and the workplace of the future. ***The Academy of Management Executive***, 12(3): 17–29.
Uzzi, B., & Gillespie, J. J. 2002. Knowledge spillover in corporate financing networks: Embeddedness and the firm's debt performance. ***Strategic Management Journal***, 23(7): 595–618.
Whiting, V. R., & Reardon, K. K. 1998. Communicating from a distance: establishing commitment in a virtual office environment. ***Academy of Management***.
Wong, S.-S., & Burton, R. M. 2000. Virtual teams: what are their characteristics, and impact on team performance? ***Computational & Mathematical organization theory***, 6(4): 339–360.

Appendix:

Table A1: Descriptive Statistics

	Variable	*Observations*	*Mean*	*Std. Dev.*	*Min*	*Max*
Projects **outside** the giant component	Δdownload	27410	20733.59	1109424	0	1.71e+08
	Δbetween	27410	-1.08e-07	1.52e-06	-.0000842	9.55e-09
	ΔCloseness	27410	-.0015933	.0067119	-.0418276	.0001101
	ΔDegree	27410	-.0322875	1.134059	-4	19
	ΔCpp	27410	.0694637	.5928583	-1	20
	ΔStage	27410	0.0444728	.2061469	0	1
	ΔStars5	27410	-.0048887	.2240058	-1	1
	years_since	27410	6.57	1.55	3.97	10.15

	Variable	*Observations*	*Mean*	*Std.Dev.*	*Min*	*Max*
Projects always in the giant component	Δdownload	10818	69818.77	2052881	0	1.98e+08
	Δbetween	10818	8.90e-07	. 0000388	-.0007162	.0024578
	ΔCloseness	10818	.0000653	.002364	-.0036971	.0200627
	ΔDegree	10818	.6863561	3.905711	-4	103
	ΔCpp	10818	.5878166	3.099011	-1	104
	ΔStage	10818	.0444629	0.2061308	0	1
	ΔStars5	10818	.0086892	.3565553	-1	1
	years_since	10818	7.34	1.59	3.97	10.16

	Variable	*Observations*	*Mean*	*Std. Dev.*	*Min*	*Max*
Projects that moved into the giant component	Δdownload	2656	54599.41	878090.8	0	3.74e+07
	Δbetween	2656	3.66e-06	.0000167	-9.03e-09	.0006224
	ΔCloseness	2656	.0297832	.004486	.0160351	.0454347
	ΔDegree	2656	1.907003	3.467798	-4	81
	ΔCpp	2656	.8524096	3.349011	-1	86
	ΔStage	2656	.1125753	0.316	0	1
	ΔStars5	2656	.1716867	.4432846	-1	1
	years_since	2656	6.34	1.60	3.97	10.11

Table A2: Correlation Among All Centrality Variables (Giant Component: N=13,474)

	ΔCpp	Δdegree	Δcloseness	Star
ΔCpp	1.00			
ΔDegree	0.53	1.00		
ΔCloseness	0.06	0.18	1.00	
Star	0.09	0.34	0.21	1.00

Table A3: Replicating Analysis in Table 2 Using All Observations

Dependent Variable: lΔdownloads	Case II: In the Giant Component	Case IIA: Always in the Giant	Case IIB: Moved into the Giant
Constant	-12.45 (-7.86)	-11.18 (-6.61)	-29.29 (-5.92)
lΔCpp	4.30 (13.25)	3.94 (11.16)	5.63 (6.71)
lΔDegree	1.10 (3.87)	0.93 (3.13)	5.26 (4.71)
lΔCloseness	0.46 (2.95)	0.34 (2.04)	1.97 (3.94)
ΔStars5	0.33 (6.03)	0.29 (4.59)	0.18 (1.52)
ΔStage	1.52 (17.55)	1.69 (15.75)	01.00 (7.13)
lyears_since	.33 (2.87)	0.45 (3.46)	-0.10 (-0.44)
Moved into giant	-0.64 (-4.19)		
# of Observations	14,939	12,251	2,688
Adjusted R-squared	0.05	0.04	0.11