Emergence of New Project Teams from Open Source Software Developer Networks: Impact of Prior Collaboration Ties[◊]

Jungpil Hahn[†], Jae Yun Moon[‡], and Chen Zhang[†]

Last Revised: July 15, 2006

ABSTRACT

Software development has traditionally been regarded as an activity that can only be effectively conducted and managed within a firm setting. However, contrary to such assertions, the open source software development (OSSD) approach, in which software developers in Internet-based communities coordinate to voluntarily contribute programming code, has recently emerged as a promising alternative to developing high quality software. Although many high profile cases of successful OSSD projects exist (e.g., Apache, OpenOffice, Emacs, PHP), the harsh reality is that the vast majority of OSS projects fail to take off and become abandoned. A commonly cited reason for the failure of OSS projects is the lack of developers in the project teams, or put differently, the inability of the software project to bring together a critical mass of developers. In this paper, we examine how OSSD project teams are formed. More specifically, we investigate whether prior collaborative ties impact OSSD team formation and developers' joining behaviors. Using software project data from real world OSSD projects, we empirically test the impact of previous collaborative ties on software team formation. Overall, we find that the existence and the amount of prior collaborative relations in the developer network do increase the probability that an OSS project will attract more developers and that a developer's prior relationships with a project initiator do increase the likelihood that a developer will join a project initiated by a past collaborator. We also explore the performance implications of early team formation behaviors. We discuss the implications of our results with respect to open source software development and software project management.

Keywords: Open source software development (OSSD), team formation, developer social networks, social ties, social capital

---

[◊] The authors are listed in alphabetical order.

[†] Krannert Graduate School of Management, Purdue University, West Lafayette IN, 47907 (J. Hahn: 1-765-494-2188, jphahn@mgmt.purdue.edu; C. Zhang: 1-765-496-6314, zhang153@mgmt.purdue.edu).

[‡] Business School, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. (852-2358-7635, jmoon@ust.hk).

1. INTRODUCTION

The creation of industrial-strength software code (or software development) has traditionally been regarded as an activity that can only be effectively conducted and managed within a firm setting. Recently however, an alternative model of software development, the open source software development (OSSD) model in which programmers in Internet-based communities collaborate to voluntarily contribute programming code has emerged as a promising approach to developing high-quality software (Raymond 2001). During the past few years, a number of open source software (OSS) products, ranging from end-user applications (e.g., Emacs and OpenOffice), programming languages (e.g., Perl and PHP) to applications supporting the Internet infrastructure (e.g., sendmail), have been widely adopted. The prominence garnered by well-known OSS projects such as the Apache Web Server and the Linux operating system kernel are testimonies to the attractiveness and viability of OSSD as an alternative to the conventional proprietary model of producing software (O'Reilly 1999, Raymond 2001).

Open source software project success depends on successfully attracting and sustaining volunteer developers and effectively coordinating their contributions. The first of these issues is the question addressed in a growing body of literature that examines factors that motivate individuals to participate in OSSD (i.e., the creation of a public good) despite the lack of monetary compensation (e.g., Hars and Qu 2002, Lerner and Tirole 2002, Roberts et al. 2006). From a public goods perspective the OSSD model is untenable because potential developers who could make high quality contributions have no incentives for doing so as they will not be able to benefit in return from others' contributions (e.g., Connolly and Thorn 1990). The "private-collective" model of innovation offers an alternative explanation of the viability of OSSD (von Hippel and von Krogh 2003). According to the "private-collective" model, a person may contribute to the creation of a public good when private benefits outweigh the potential private losses of revealing one's innovation and when the private benefits of contribution outweigh those available to free riders (i.e., non-contributors). The OSS literature points to different types of benefits that motivate OSS developers. Hars and Qu (2002) identify both intrinsic motivations such as altruism and extrinsic motivations such as direct compensation. Another study surveys the motivations of the contributors to a large OSS project and finds that participation is mainly driven by developers' group identification, by the possibility of improving their own software, and by their tolerance of the required time investments for contributing to the project (Hertel et al. 2003). Lakhani and Wolf (2005) identify enjoyment-based intrinsic motivation, user need, and learning as the most pervasive drivers of developer participation. Other possible explanations for developers' participation in OSS projects include motivations related to career concerns and ego gratification (Lerner and Tirole 2002). Overall, the prior literature suggests that developers participate in OSSD mainly because of intrinsic factors such as enjoyment and extrinsic factors such as career advancement.

The success of projects such as Linux and Apache has resulted in a dramatic increase in the number of open source software projects launched by both commercial and non-commercial actors. However, despite increased understanding of the reasons behind the impressive success of some OSSD projects, the harsh reality is that the vast majority of OSS projects fail to take off and become abandoned. One of the main reasons cited for the failure of OSS projects is the lack of developers in the project teams, or the inability of the project to bring together a critical mass of developers (Lerner and Tirole 2001, O'Reilly 1999, von Krogh et al. 2003). Since it is typically the case that OSSD projects do not provide monetary rewards for developers' contributions, many OSSD projects are under-staffed and consequently are not well-equipped to deal with the complexity in software development (von Krogh et al. 2003). While the prior literature makes a convincing argument for why developers would choose to participate in OSS projects (despite the lack of direct compensation), it is still unclear why developers choose to join one particular project over other similar alternative projects.[1] In most cases the formation of an OSSD project team is a dynamic, self-organizing process in which developers voluntarily choose to become members of the project. The collective results from the research on OSS developer motivations suggest that developers will be most likely to join OSSD projects that they perceive will provide the

[1] But see Stewart et al. (2006) for a notable recent exception that examined OSS project characteristics that affect developer motivations to contribute to a particular OSS project. The focus of our study is

greatest opportunity for realizing private benefits such as learning, reputation or enjoyment. Successful OSSD projects that result in a large user and developer base are most likely to provide such opportunities. While mature OSSD projects with large user bases are likely to provide opportunities for learning and personal enjoyment, they may provide fewer opportunities to realize reputation benefits due to the relative stabilization of core developer status and code base within these groups. How then do developers evaluate projects at the early stage when choosing which OSSD project to contribute to? Familiarity bred from past interactions and work relationships has been identified as one important factor in work group formation (Hinds et al. 2000, Zander and Havelin 1960). The goal of this paper is to explore how prior project collaborative ties affect developer choice of newly-initiated open source software projects to contribute to.[2]

Understanding how OSS project teams are formed is important for several reasons. Research on small groups has shown that group composition is an important determinant of group performance through its impact on group cohesion and coordination among other factors (Beal et al. 2003, Gruenfeld et al. 1996, Levine and Moreland 1990). Member composition can also affect software development team performance through its impact on administrative and expertise coordination effectiveness (Faraj and Sproull 2000b). Software development team performance is also affected by the programming skills and application domain experiences of team members (Boehm 1987, Curtis et al. 1988). While organizational software development teams are formed by managers based on developer skills and experiences, in OSSD teams the formation of the team is not controlled by project leaders. Hence, in order to understand and solve the key problems related to staffing and project performance, it is important to understand the dynamics of OSS team formation – how OSS projects attract developers to join the development team

---

slightly different in that we examine not projects that are already active, but newly initiated projects in order to understand the initial phase of project team formation.

[2] We focus on developer choice of newly initiated open source projects to eliminate a rival explanation for group formation – namely that of preferential attachment (Barabasi and Albert 1999) in which developers base their choice on the expressed preferences of other developers whereby developer project choice becomes dominated by perceived popularity and success of groups. Early marshalling of

and how developers choose software project teams to contribute to. A secondary goal of this paper is to examine the subsequent impact of the OSS project team composition on project performance and outcomes.

In this paper, we undertake an empirical examination of the formation of OSSD project teams from a social network perspective. The OSSD community can be regarded a complex collaborative social network endowed with social capital. Just as the social position of an individual within a network of peers influences his/her career advancement opportunities (Burt 1992), or as the social position of a firm within a network of organizations influences its alliance strategies and consequent outcomes (Gulati 1995, Powell et al. 1996), we theorize that relations forged during past collaborations in open source software development will impact how OSSD project teams take form. The original Linux operating system kernel development group for example was not formed out of a social vacuum. Linus Torvalds, the project initiator, had been an active member in a related community of minix programmers and it was out of this community that the first volunteer developers emerged to form the Linux kernel developer team (Moon and Sproull 2002). A recent study that examined how subprojects evolved within the Apache project found that "most new project include at least one large group migrated from another project" (Weiss et al. 2006, p. 29). However, despite the apparent relevance and importance of social capital in OSSD, only a relatively few studies have examined its impact on developers' team formation behaviors from a social network perspective. In this paper, we ask whether the existence and amount of prior collaborative ties of an OSS project initiator helps in attracting additional developers. The remainder of this paper is organized as follows. In the next section, we present our theoretical background and develop our research hypotheses. Then we outline the empirical research methodology and present the results. We conclude by discussing the implications, contributions and directions for future research.

---

developer interest is thus what becomes critical for ensuring open source software project success and the focus of our paper.

2. THEORETICAL FRAMEWORK AND HYPOTHESES

2.1. A Social Network Perspective of Open Source Software Development

Social network analysis aims to understand the relationships between people, groups, organizations, and other types of social entities (Granovetter 1973, Wasserman and Galaskiewicz 1994, Wellman and Berkowitz 1998), and has been used extensively in fields such as sociology (Cook and Whitmeyer 1992, Wasserman and Galaskiewicz 1994) and management (Borgatti and Foster 2003, Tsai 2001) among others (Huang and DeSanctis 2005, Singh 2005). A social network is modeled as a graph with nodes representing the individual actors in the network and ties representing the relationships between the actors.

In a social network the actors maintain a tie by exchanging either tangible or intangible resources such as information, goods and services, and financial support. The strength of a tie varies depending on a number of factors. Granovetter (1973) distinguishes between strong and weak ties and asserts that tie strength depends on the amount of time, the emotional intensity, the intimacy, and the reciprocal services associated with the relationship. Strong ties are characterized by a sense of special relationship, an interest in frequent interactions, and a sense of mutuality of the relationship (Walker et al. 1994). In contrast, weak ties are maintained infrequently or indirectly between the actors who belong to different social clusters. Both strong ties and weak ties play an important and differential role in a social network. Strong ties maintain and promote trust and collaboration whereas weak ties enable actors to access resources and information that are unavailable in their immediate social circles (Burt 1992, Granovetter 1973).

Although it has been recognized early on that OSSD has become a significant social phenomenon and that OSS developers and users form a complex social network via various electronic communication channels on the Internet (von Hippel and von Krogh 2003), few researchers have examined this phenomenon from a social network perspective. Madey, Freeh, and Tynan (2002) conducted one of the first empirical investigations of the open source movement from this perspective by modeling OSS projects as a collaborative social network and found that the OSSD community can be modeled as a self-organizing social network. Others propose the methodology of applying social network analysis to data gathered from CVS code repositories of OSS projects (Lopez-Fernandez et al. 2004). Xu, Gao, Christley, and Madey (2005) explored some social network properties in the open source community to identify patterns of collaborations. However, these earlier studies tend to be highly technical and mainly investigate the network properties of the OSSD community, offering limited theoretical and practical contributions. The work most similar to our research is done by Ducheneaut (2005) who examined the socialization process of newcomers over time as a learning process and a political process by analyzing the developer activities in a large OSS project.

In the context of OSSD, when deciding whether to join a project, in addition to the previously cited motivational factors, factors relating to the collaborative relationships between developers may become important. For example, a developer may be concerned about issues related to coordination and communication with other team members. In general, when forming teams people prefer to work with those with whom they have worked in the past because of the reduced uncertainty stemming from familiarity based on previous collaborative experiences (Hinds et al. 2000). Familiarity bred from preexisting working relations with others can also facilitate the newcomer's socialization process. Hence, we focus on the impact of prior collaborative ties among developers as a potential driver behind developer joining behavior and project team formation; in short, we examine how collaborative tie networks impact the project teams that emerge.

2.2. Perspectives on How Prior Collaborative Ties Affect Project Team Formation

Conventionally, project teams in organizations are strategically formed by a manager assigning individuals to a team based on certain characteristics such as expertise and personality. An alternative approach is driven by team members' self-selection into teams. Similarly, in OSSD, some project initiators may formally recruit developers[3] (e.g., by broadcasting position openings and required qualifications to the entire community), or alternatively developers may voluntarily join a project team or be invited to participate in a project team by its existing members. In this section we integrate emergent findings from research on motivations of OSS developers with theoretical perspectives derived from organizational behavior, social psychology and sociology that have examined the mechanisms and processes that determine the formation of naturally occurring groups both within and outside organizations (e.g., Ruef et al. 2003).

Research on group formation – be it work or social, self-organized or prescribed – indicates that group formation is a result of the deliberate, strategic decisions of individuals who either self-select or assign others to a group with the purpose of satisfying individual and group objectives (Owens et al. 1998). OSS developers will choose projects that afford them ample opportunities to realize the expected benefits of participation. Benefit realization is contingent on OSS project success. OSS developers thus may rely on personal experiences with the project initiator to judge the likelihood of the successful outcome of the new project. Prior research suggests that people are more likely to work together when they have prior social ties (McClelland et al. 1953, Schachter 1959). Moreover, teams consisting of individuals with preexisting relationships have been shown to solve complex problems better than teams of strangers because they are able to pool information more efficiently (Gruenfeld et al. 1996). Software development teams composed of members with prior joint project experience may be more effective in coordinating programmers' distributed expertise because they have developed knowledge of 'who knows what' (Moreland 1999). In the open source software development context in particular, due to the lack of opportunities for face-to-face contact, developers face greater barriers to effective communication and coordination and are thus more likely to be concerned about these issues. Previous collaborative relations

[3] Interestingly, the extent of *active* recruiting is surprisingly low (von Hippel and von Krogh 2003). For example, there are only on average 200 position openings posted on SourceForge.net at any given time. When we consider that there are over 100,000 OSS projects hosted on SourceForge.net, this number is quite inconsequential. (These numbers are based on SourceForge.net statistics gathered in November 2005.)

with existing members of a project can mitigate concerns regarding communication and coordination difficulties due to the shared context accrued from prior interactions (Hinds et al. 2000, Moreland 1999). OSS developers will more likely be attracted to projects that are initiated by developers with whom they are familiar based on prior project collaborations. Moreover, the more collaborative ties the initiators have, the larger will be the pool of potential developers. In other words, given that an initiator has developed a network of relations through participation in past OSSD projects, the denser this network of collaborative ties, the higher the visibility of a new project launched by the initiator and the more likely the project initiator will be to attract additional developer resources. Consequently, these projects will be able to attract or invite others into the development team more easily. Therefore, we propose the following hypothesis regarding the impact of the amount of preexisting strong ties in a project:

H1: (Project Level) *The number of project initiator's collaborative ties with OSS developers in the network is positively associated with both (a) the probability of having other developers join the project team, and (b) the number of developers who join the project team.*

H2: (Developer Level: Developer-Project Dyad) *The probability that a developer joins a project is positively related to the existence of a prior collaborative tie with the initiator.*

Not all prior collaborative experiences are positive. Some projects fail and may result in forking of the code due to conflict between project members. Because OSS developers are motivated to maximize the chance that they can gain the expected benefits from OSS project participation, they are more likely to choose the same coordinator only if the past collaborations have been successful both in terms of final project outcome as well as in terms of the quality of the coordination and collaboration process. Research on group formation in laboratory and field study contexts has found that people are attracted to groups when their prior experiences with key group members have been positive and successful (Hinds et al. 2000, Zander and Havelin 1960). In addition, positive collaboration experiences may have created implicit obligations for future exchange due to the benefits that the developer derived from association with the project initiator in the past, in particular if the past collaboration experience was one in which the current project initiator contributed substantially to the developer's past project . Thus, we hypothesize:

H3: (Developer Level: Developer-Project Dyad): *The quality and strength of the collaborative tie will moderate the relationship between the existence of a collaborative tie and the likelihood that the developer will join the project. More specifically, the likelihood that a developer will join a project will be greater when they share a positive, strong collaborative tie with the project leader than when the tie is weak.*

In addition to impressions formed through direct interaction and collaboration with the project initiator, OSS developers may use indirect experiences of other developers in the collaboration network to judge the likelihood of project success. Prior research suggests that potential OSS developers will prefer participating in projects initiated by people who are perceived to have higher status on average. Such a group formation mechanism is also due to the belief that high status individuals are more competent and hence have a greater likelihood of initiating a successful project (Stewart 2005, Thye 2000). Because developers are more likely to join an OSSD project if they perceive that their contributions are instrumental for successful project outcome (Karau and Williams 2000), projects initiated by developers of higher perceived status may be more likely to have additional team members because of the increase in perceived likelihood of new member contributions leading to valuable outcomes. In addition, because of the reputation-conferring benefits of association with high-status participants in the network (Stewart 2005), OSS developers motivated by reputation will be more likely to join projects that are initiated by members with high perceived status in the community. Status perceptions are influenced by the extent to which the project initiator is perceived as being embedded within a dense network of developer collaborative ties, that is, the relative perceived centrality of the project initiator within the OSS developer network. Thus, we hypothesize

H4: (Developer Level: Developer-Project Dyad): *The probability that a developer joins a project is greater when the project initiator has a greater amount of preexisting collaborative ties with the open source software developer network.*

3. RESEARCH METHODS

3.1. Data Collection and Measures

Project and developer data were obtained from the dump of SourceForge.net's project databases currently hosted at the University of Notre Dame (http://www.nd.edu/~oss/). As the largest repository of open source applications on the Internet, SourceForge.net currently provides free hosting to more than 100,000 projects and more than 1,100,000 subscribers. It also offers a variety of services to hosted projects, including site hosting, mailing lists, bug tracking, message boards, file archiving, and other project management tools. SourceForge.net has been an attractive source of data for many researchers studying open source software mainly due to the abundance of publicly accessible data (Howison and Crowston 2004).

We selected all public OSS projects newly registered on Sourceforge.net between September 13, 2005 and October 14, 2005 ($N = 1780$). This was the sample for hypothesis testing at the project level. We revisited these projects on November 21, 2005 to capture the developers who had subsequently joined the project in the first 1 to 2 months. This process enables us to distinguish between the initiator and the developers who subsequently joined. Further, in order to identify the previous collaborative ties of the developers, we collected data on other projects that each developer had participated in prior to his/her joining the focal project to identify his/her past collaborators. Based on this data, we constructed affiliation matrices of developers and projects that depict the existence of the relationship ties between developers.

At the developer-project dyad level, hypothesis testing was conducted using a subset of the sample described above. Sample selection was based on availability of project and developer information required to operationalize measures of fit between developers' technical skills and project technical requirements. (The specific variables that we constructed for hypothesis testing are described in more detail below.) First, we included only those projects that explicitly defined technical details such as programming language, domain of software and operating system platform. Second, we restricted the sample of SourceForge.net developers to those who had participated in at least one project by October

2005. This is because since only 2.3% of developers make their technical skill profiles accessible on SourceForge.net, we inferred the developers' technical skills from their past project experience. The final dataset used for hypothesis testing at the developer-project dyad level consists of 938 projects and 173,523 developers.

Dependent Variables

The final outcome of interest at the project level is whether and to what extent a project has successfully attracted developers to join. The former is measured as a binary variable to indicate developer joining of the project within the first one or two months of project inception (*DeveloperJoin*); the latter is operationalized as the number of developers joining a project within the target time frame (*NumJoiningDev*). At the developer-project dyad level the dependent variable is a binary indicator that captures whether a developer joins a particular project within the first 2 months of project initiation (*Join*).

Independent Variables

To test the effects of prior collaborative ties on the formation of OSSD teams, we first constructed a measure of a project initiator's social capital and status based on the number of developers in the open source software development network with whom he/she has had previous collaborative ties with (*InitiatorTieAmount*) prior to project inception.

At the developer-project dyad level additional measures were constructed to represent the presence and strength of prior collaborative ties for each developer and project initiator pair. The existence of a past collaborative tie between a developer and a project initiator was operationalized as a binary indicator variable (*HasTie*). The quality and strength of the tie between the developer and project initiator may vary depending on the nature of the past collaborative experiences, that is, depending on the outcome quality of the project as well as the coordination process quality. In order to operationalize tie strength and quality, we characterized the nature of the past collaborations based on Granovetter's (1973) distinction between strong and weak ties and on measures of OSS project success (Crowston et al. 2003). For each developer-project initiator pair we constructed seven measures of joint project activity. The extent to which past collaborative experiences had been successful was measured by the average amount of code released (*AverageBytes*), the number of code downloads by users (*AverageDownloads*), whether or not the project resulted in a successful release of the working program code (*AverageHasRelease*), and the development status of the project (*AverageDevelopmentStatus*). The intimacy and amount of time spent in cultivating the tie was measured through the number and duration of the projects they collaborated on (*NumCollaborations, AverageDuration*). Because we expect project administrators to interact more frequently in order to coordinate the project we also tracked whether the developer and project initiator had shared project administration responsibilities in prior projects (*BothAdminRole*). We conducted exploratory factor analysis to determine the underlying latent constructs to measure the strength of a past collaborative tie. Two factors emerged from the analysis. The first factor, which we call *TieStrengthProduct*, represents tie strength based on whether or not the *outcome* of the past collaboration between the developer and the project initiator was positive, that is whether the project was successful as measured through number of download among other factors. The second factor, *TieStrengthProcess*, represents tie strength that is dependent on the *process* of coordination such as whether or not the developer collaborated as administrator with the project initiator in past collaborations. The measures were constructed using factor scores based on the factor loadings shown in Table 1.

Place **Table 1** Here

Control Variables

We controlled for project initiator-related characteristics that may have an impact on developers' joining decisions. It is likely that developers with prior open source project experience will have superior knowledge of OSS development and management processes, increasing the likelihood that the project outcome will be successful. Therefore, we measured the experience of project initiators using absolute participation duration at SourceForge.net (*InitiatorExperience_t*) as well as the number of projects in which the initiator has participated in the past (*InitiatorExperience_p)*. We captured additional developer level characteristics that would affect developers' decisions to join a particular project. We measured the experience of developers in terms of number of projects (*DeveloperExperience_p*) they participated in and total participation duration (*DeveloperExperience_t*) at SourceForge.net. We also considered the technical fit between developers' skills and the focal project's requirements in order to control for the impact of developer expertise and interest on project selection. These were captured in three variables that reflected whether the technical details in terms of topic (*MatchTopic*)*,* programming language (*MatchProgLang),* and application platform (*MatchOS*) of any of the projects in which the developer had participated previously matched the details of the new project.

We also controlled for other project attributes that would influence developer joining decisions. These included attributes of the project that would affect the visibility of the project to potential developers such as whether the project has been included in the OSSD community SourceForge software map *(TroveDefined),* the lifetime of the project (*Duration*) and the popularity of the project application domain measured as a proportion of developers for the top 1000 projects who work on this domain (*TopicPopularity*). Other project attributes we measured are the level of details available in the project description that would facilitate information gathering required for making a joining decision (*DescDetail*) and whether the project is set up to accept donations from users (*AcceptDonation*). The summary of the measures computed for empirical analysis are shown in Table 2.

Place **Table 2** Here

3.2. Analytical Procedures

Project Level Analyses.

We tested the impact of project initiators' prior collaborative ties and development experience on the binary measure of project success in attracting developers (*DeveloperJoin*) using a logistic regression framework. In other words, we estimate the parameters for a logistic regression model of the form:

$$\Pr(y = 1 \mid x = x_i) = \text{logit}(x_i\beta) = \frac{1}{1 + e^{-x_i\beta}}$$

where y is the event that at least one developer has joined the project (i.e., *DeveloperJoin*), x_i is the vector of covariate values and β is the vector of parameters to estimate.

We use negative binomial regression to test the hypothesized effects of collaborative ties and development experience on *how* successful projects are in attracting developers (i.e., *NumJoiningDev*).[4]

Hypothesis Testing at Developer-Project Dyad Level.

To test hypotheses at the developer-project dyad level, we adopted the technique of choice-based sampling or endogenous stratified sampling (King and Zeng 2001, Manski and Lerman 1977).[5] The strategy is to choose a fraction of the developer-project dyads representing the joining event and to choose a much smaller fraction of the non-event pairs. We used our sample of projects (N = 938) and selected all developers who have joined these projects as the event sample. In addition, we matched each dyad in the event sample with six control dyads as the control sample while ensuring that control sample has similar

[4] Although Poisson regression is a common analysis technique when the dependent variable has only non-negative integer values, our data suffers from problems of overdispersion (Deviance/DF = 1.914, Pearson χ^2/DF = 4.246; LR statistic = 1316.479, $p < 0.01$) mainly because many observations in our dataset have a value of zero for the count variable. Therefore, we test our hypothesis using a negative binomial regression model that allows for correction of overdispersion (Allison 1999).

[5] For our sample, conventional logistic regression approach with random sampling is impractical due to the rarity of a developer's project joining event. For instance, with approximately 1,000 sample projects and 170,000 sample developers, there would be over 170 million (i.e., 1,000 × 170,000) developer-project dyads in total. However, of those possible dyads, there are only a very small percentage of dyads

(and/or dissimilar) characteristics as the event dyads. In particular we controlled for the match between project requirement and developer skills (e.g., software topic, programming language, operating system) as well as the existence of prior collaborative social ties. In addition, two random dyads are selected for each event dyad. The choice-based sampling procedure produced a sample of approximately 3,800 dyads.

Corresponding to the choice-based sampling technique, we adopted the weighted exogenous sampling maximum-likelihood (WESML) estimator (Manski and Lerman 1977) as a validated approach adopted in prior literature (e.g., Singh 2005). The WESML estimator is calculated by maximizing the weighted pseudo-likelihood function that weighs each observation in the sample with the number of population observations that it represents. For example, the weight of a sample dyad that represents 100 potential dyads in the entire population is 10 times the weight of a sample dyad that represents 10 population dyads[6]. In addition, because the same developer may be included in multiple project developer dyads, we calculated the standard errors without assuming independent errors among observations.

In order to investigate the robustness of the estimation with respect to the choice-based sampling procedure, we drew 1000 bootstrap choice-based samples to derive the bootstrap mean and the confidence intervals for each parameter estimate.

4. RESULTS

4.1. Project Level Impact of Project Initiator's Prior Collaborative Ties

The project sample descriptive statistics are shown in Table 3. Table 4 presents the pairwise correlations of the measures. The correlation between *InitiatorTieAmount* and *InitiatorExperience_p* is moderately high ($\rho = 0.642$, $p < 0.001$). However, further collinearity diagnostics show that the

---

representing the event that a developer joined a project. Thus, pure random sampling from all possible dyads would make the sample size impractically large and lead to biased statistical estimation.

[6] For more technical details on WESML refer to King and Zeng (2001).

tolerances of all predictor variables are above 0.4 and that the highest variance inflation factor value is 2.123, indicating that multicollinearity is not a major concern in the analysis.

Place **Tables 3** and **4** Here

We hypothesized that the more collaborative ties project initiators had with OSS developers in the network the more likely developers would be to join the project (H1a). We used the following logistic regression model to test our hypotheses:

$$\begin{aligned}\operatorname{logit}\left(\Pr(y=1)\right) = \alpha + \beta_1 InitiatorTieAmount + \beta_2 InitiatorExperience\_t \\ + \beta_3 InitiatorExperience\_p + \beta_4 DescLength + \beta_5 TroveDefined \\ + \beta_6 AcceptDonation + \beta_7 TopicPopularity + \beta_8 Duration + \varepsilon\end{aligned}$$

A positive and significant estimate of parameter β_1 would indicate that the probability of other developers becoming members of a project is positively related to the amount of collaborative work relations the initiator has maintained in the network. The results of the logistic regression are presented in Table 5 (Model 1). The model fits with the data moderately well (likelihood ratio $\chi^2 = 54.71$, $p < 0.01$; Hosmer-Lemeshow statistic = 6.43, $p = 0.60$). The variable *InitiatorTieAmount* has a significant and positive effect on the likelihood of developers joining ($\beta_1 = 0.516$, $p < 0.01$). The results suggest that an additional tie for an initiator increases the likelihood of at least one developer joining the project team by 67.6%. Given that on average an initiator has had prior relationships with approximately 3 other developers in our sample, this would on average amount to a doubling of the likelihood. Thus, projects whose initiators have more collaborative ties with the developer network are more likely to attract additional developers than those whose initiators have fewer ties (H1a is supported).

Next, we examined the factors that may impact the number of additional developers entering a project team by estimating the parameters for the following negative binomial regression model (H1b):

$$\log E(y) = \ln(Duration) + \alpha + \beta_1 InitiatorTieAmount + \beta_2 InitiatorExperience\_t$$
$$+\beta_3 InitiatorExperience\_p + \beta_4 DescLength + \beta_5 TroveDefined$$
$$+\beta_6 AcceptDonation + \beta_7 TopicPopularity + \sigma\varepsilon$$

The results are summarized in Table 5 (Model 2). The estimates for *InitiatorTieAmount* is significant and positive ($\beta_1 = 0.848$, $p < 0.01$), indicating that projects initiated by the developers with more collaborative ties tend to attract more developers than those initiated by the developers with fewer ties (H1b is supported). Overall, the results of the negative binomial regression are quite consistent with the results obtained from the logistic regression. The results also indicate that whether the project has defined its technical properties positively influences both the probability of attracting additional developers and the number of additional developers. However, the initiator's experience in terms of time spent in the community and the number of projects he/she contributed to was significantly negatively related to project likelihood of attracting volunteer developers, a finding that may seem counter-intuitive. One possible explanation may be that developers in the OSS community support newcomers by joining their projects and at the same time expand their existing social relations in the network.

Place **Table 5** Here

In summary, we find that at the project level the quantity of an initiator's prior collaborative ties positively influences not only the probability of additional developers joining the project but also the number of such developers.

4.2. Developer-Project Dyad Level Analysis of the Impact of Prior Collaborative Ties with Project Initiator on Developer Project Joining Decisions

Table 6 summarizes the descriptive statistics and Table 7 presents the pairwise correlations of the measures for the developer-project dyad sample. The highest correlation among the independent variables is between *InitiatorTieAmount* and *InitiatorExperience_p* ($\rho = 0.652$, $p < 0.001$).

Place **Tables 6** and **7** Here

At the developer-project dyad level, we hypothesized that the collaborative tie between the developer and the project initiator (H2, β_1), the strength and quality of the collaborative tie (H3, β_2, β_3) and the number of project initiator's collaborative ties (H4, β_4) would have a positive impact on developer decisions to join a project. We used WESML to estimate the parameters for the following logistic regression model:

$$\begin{aligned}\text{logit}\left(\Pr(y=1)\right) = \alpha &+ \beta_1 HasTie + \beta_2 HasTie * TieStrengthProduct + \beta_3 HasTie * TieStrengthProcess \\ &+ \beta_4 InitiatorTieAmount + \beta_5 InitiatorExperience\_t + \beta_6 InitiatorExperience\_p \\ &+ \beta_7 DeveloperExperience\_t + \beta_8 DeveloperExperience\_p \\ &+ \beta_9 MatchTopic + \beta_{10} MatchProgLang + \beta_{11} MatchOS + \beta_{12} Duration \\ &+ \beta_{13} DescDetail + \beta_{14} AcceptDonation + \beta_{15} TopicPopularity + \varepsilon\end{aligned}$$

The results of the logistic regression are presented in Table 8 (Model 4). The variable *HasTie* has a significantly positive impact on the likelihood of developer joining ($\beta_1 = 7.244$, $p < 0.01$), suggesting that a developer is far more likely to join a project that has been initiated by a past collaborator whom he/she is familiar with (H2 is supported). Furthermore, the interaction term between *HasTie* and *TieStrengthProcess* has a positive and significant estimate whereas the interaction term between *HasTie* and *TieStrengthProduct* is not significant, indicating that the past collaboration process itself has a greater moderating influence on a developer's future joining decisions than the successful production of software (H3 is partially supported). The parameter estimate for *InitiatorTieAmount* is significant and positive (H4 is supported). A developer is more likely to join a project whose initiator has more ties. Lastly, control variables capturing the experience level of the project submitter and the developer negatively influenced project joining decisions, indicating that OSS developers prefer to join projects initiated by newcomers when deciding. In summary, the results indicate that both the existence and the process-related quality and strength of collaborative ties between the developer and the project originator positively impact the likelihood of the developer joining the project.

4.3. Post-Hoc Analyses of Project Joining Decisions

We performed additional exploratory analyses to investigate whether the same pattern of results can be observed in terms of the joining decision regardless of the experience level of developers. Experienced developers may have a different set of motivations for OSSD project participation and thus may employ a different decision calculus when deliberating the choice of project to join. We thus divided the sample based on developers' past project experience and performed the logistic regression analysis. Model 5 shows the results for the developer-project dyads in which the developer has participated in one project in the past (i.e., less experienced developers) whereas Model 6 is associated with the dyads in which the developer has worked on more than one projects (i.e., more experienced developers). Results are presented in Table 8.

The parameter estimates for *HasTie* and *HasTie*×*TieStrengthProcess* are still significantly positive for experienced developers (Model 6). However, they are no longer significant predictors of joining behaviors for inexperienced developers (Model 5). In addition, there are some significant differences in the parameter estimates between the two models. For example, estimates for *MatchTopic* and *MatchOS* are both negatively significant in Model 5, suggesting the presence of possible learning effect whereas those for *MatchTopic* and *MatchProgLang* are both significantly positive in Model 6, suggesting that the experienced developers tend to join projects related to the topic domains and programming languages that they are familiar with. Overall the results indicate that the factors influencing experienced developers' joining decisions are quite different from those influencing inexperienced developers' decisions. Inexperienced developers seem to choose which project to join mainly based on potential learning benefits whereas experienced developers tend to make the joining decision based on their past relations with project initiators as well as their personal interests in software domains and programming languages.

Place **Table 8** Here

---

5. DISCUSSION

In this study we investigated the role of prior collaborative ties in OSSD team formation. This research fills a gap in the open source literature by conducting an empirical investigation of the role of social relations on project team formation behavior. Furthermore, the adoption of a social network perspective, which has received little attention in the OSS literature, yielded some interesting results with respect to the interactions among OSS developers. We examined how relationships developed through past collaborations in the open source software community in turn affect the formation of new open source software project teams. Specifically, we examined whether prior collaborative ties with developers in the open source software development community in which the project is embedded impact the probability of an OSS project to attract more developers. Our results supported our main thesis that the network of collaborative ties in OSSD networks plays an important role in new project team formation processes. We found that overall the existence and number of prior collaborative ties increase the probability that developers join a project. Moreover, collaborative tie strength moderated this effect, with developers even more likely to join a project initiated by a developer with whom they have had high quality, positive experiences in the past. One obvious practical implication of our study is that commercial actors interested in rallying OSS developers to contribute to their projects may benefit from actively cultivating relationships with existing developers by among other means actively contributing in existing OSSD projects.

There were however some unexpected patterns of developer joining behaviors that warrant further discussion. In all our analysis, we found a significant negative impact of project initiator experience on developer joining decisions, indicating that developers are more likely to join projects initiated by people with less OSSD experience. Less experienced OSS developers were also more likely to join projects that allowed them to gain experience contributing to projects that made use of different programming languages and developed applications for new application domains. It is possible that these differences in antecedents of project joining decisions may reflect differences in underlying participation motivations for experienced and novice OSS developers. Novice OSS developers were also not affected by past collaboration with a project initiator in deciding which project to join. However our findings also suggest that as OSS developers gain more experience they will be more likely to reinforce collaborative ties with a few select group of developers in the network when choosing from new OSSD projects to contribute to. Then, what are the project performance implications of such preferential attachment processes? Specifically, the results of this study lead us to pose two important follow-up questions. First, do projects started by a developer with more collaborative ties in the OSS developer network perform better? Second, does the developer joining behavior in a project's early stage impact its performance? We conducted additional exploratory analyses to provide some initial insights into these issues.

We identified three measures of project performance – *HasRelease*, *ProjectActivityScore*, and *DevelopmentStatus* that respectively measured project output, level of development activity including communication, and project administrator's self-reported project development phase,[7] which we assessed 7-8 months after project initiation with the June 2006 dump of the SourceForge.net project database. After discarding projects that were no longer active, we analyzed the effect of group composition as well

[7] See Crowston et al. (2003) for other measures of OSS performance.

as the number of developers who had joined the project in the first 1-2 months of project inception (*NumJoiningDev*) on the performance of these 1775 projects (see Table 9).[8]

Place **Table 9** Here

A comparison of these results shows that the role of a project initiator's social ties does not seem to be significantly important, especially with respect to whether the project was able to quickly produce a release or whether the project has progressed to a more mature stage of development. Perhaps it is too early to see this. However, *NumJoiningDev* is a significant predictor of two performance measures (*ProjectActivityScore* and *DevelopmentStatus)*. Given our earlier results that *InitiatorTieAmount* does favorably influence the developer joining behavior within a project's first 1 to 2 months, we propose that although *InitiatorTieAmount* does not directly impact early project performance it may does so indirectly by enabling early process efficiencies. Another interesting finding is that the project performance in its first 7 months seems to depend on the initiator's experience as well as how well the project is defined.

However, because of the relatively short period between a project's start and our performance data collection, the above results should be interpreted with caution. The performance data were taken from the latest snapshot of the projects. A higher performance in our dataset is unlikely to guarantee the definitive future success of the project. Only time will tell. Nonetheless, our exploratory study in this section can shed some light on the factors that may have performance implications during the early phase of the project.

Our paper provides empirical evidence for one mechanism that could explain the reason behind the preferential attachment processes leading to the formation of scale-free networks that are prevalent in digital and social networks (Barabasi and Albert 1999). OSS developers choose to continue collaborating

[8] These initial exploratory results are presented for discussion purposes. Given the short period of time since the inception of the projects and the assessment of those projects' performance (i.e., Sept-Oct 2005

with people with whom they have had positive project experiences. These findings are consistent with research in the formation of founding teams for new firms that finds a tendency for entrepreneurs to prefer people with whom they already have strong ties (Ruef et al. 2003). We extend on this line of research and provide some preliminary results regarding the subsequent performance implications of teams formed of homogeneous, like-minded members. Although further study is needed to fully assess the performance implications of homogeneous group composition, we do find that overall most projects have survived 8 months into their initiation.

The findings from our exploratory study contribute to an enriched understanding of the evolution and growth of open source software projects by providing one explanation of the process through which new projects attract new developers. Critical mass theory and resource dependence theory perspectives indicate that subsequent participants will join a project only if its perceived value is high. In other words, projects that fail to attract developer interest in the first phase of development will fail to attract developers in subsequent development phases. Therefore understanding the factors that affect the decision of the first core group of developers is essential for ensuring the success of open source software projects. Although a growing body of research is beginning to explore the motivations of developers contributing to the open source software community, little research to date has attempted to understand the dynamics of how developers choose which of the myriad possible projects available to contribute to. A variety of factors may affect the choice of project team to join. Both critical mass theory and expectancy value theories suggest that developers would be influenced by project size since size is a highly visible indicator of the probability of successful outcome and value of the group (Karau and Williams 2000, Markus 1987). However these theories do not explain the decision of developers who choose to join newly formed projects where project size may be less salient.

So, why do prior collaborative ties have such a profound effect on developers' project joining decisions? The results of this research suggest that it is not only the perceived expected benefits of

---

through June 2006), these results are not meant to be used for drawing any definitive conclusions.

joining the project that is salient. In addition to such perceived benefits, developers seem to be concerned about the process of realizing those potential benefits. For instance, most of the motivations, as prescribed by the prior literature, seem to focus on extrinsic and intrinsic benefits. Extrinsic benefits are benefits derived from the outcome (e.g., development of software specific to one's needs, increase in reputation after participating in a successful software project, learning effects etc.); whereas intrinsic benefits relate to those that are attainable by virtue of participation in the project itself (e.g., enjoyment, affiliation, community identification etc.). The role of prior collaborative ties seems to be related to reducing uncertainties in the process of project participation. In fact, software development is not only a production process but also a social process that heavily involves inter-personal communication and coordination (Curtis et al. 1988, Robey and Newman 1996, Sawyer et al. 1997, Sawyer and Guinan 1998). Moreover, in the OSSD context, the difficulty inherent in the social process becomes even more accentuated than in traditional software projects since members of an OSSD project are typically from geographically dispersed locations, have diverse cultural backgrounds, and have limited (if not any) face-to-face interactions. From the developer's perspective, a large amount of uncertainty exists with regard to how smooth and how efficient it will be to interact with the other members of a project. Hence, when deciding whether to become a team member of an OSS project, in addition to the motivational factors of potential extrinsic and intrinsic benefits, a developer will also be concerned about the potential difficulty related to coordinating and communicating with other team members that will affect the likelihood of *realizing* these benefits.

Another plausible explanation may be that OSS developers, especially more experienced ones, may also prefer projects initiated by people with whom they share a collaborative tie due to a sense of belonging to the same group. A survey of Linux developers found that active developers identified strongly not only with the Linux community broadly but also with the specific subsystem team (Hertel et al. 2003). This sense of group identity will lead OSS developers to evaluate the projects initiated by those they perceive to be in-group members more favorably than projects initiated by out-group members (Hogg and Abrams 1988).

While the current study provides some answers to initial questions, it opens up a series of interesting questions that merit additional research. The tendency for OSS developers to continue to collaborate with people with whom they are familiar raises an interesting question of whether the open source software model is effective in producing truly innovative software products. Ruef et al. (2003) go so far as to consider strong tie networks as *constraints* imposed on entrepreneurial founding team formation that leads to founding of 'sub-optimal' teams in terms of functional diversity. The organizational teams literature overall suggests that the heterogeneity of group member composition is positively related to group performance, in particular for creative, intellective tasks (Guzzo and Dickson 1996). Ancona and Caldwell (1992) for instance found that groups that were more heterogeneous in terms of member expertise were more likely to generate innovative products. A cursory examination of the OSSD arena may lead one to raise concerns regarding the current state of affairs and to design interventions that may affect team formation decisions. While there has not yet been any formal analysis of the innovativeness of open source software projects in general, some of the most successful open source software projects to date are software that attempted to offer a free and open alternative of proprietary code (e.g., Linux) or projects that were spawned off as open source software for various reasons (e.g., Apache). It seems that OSSD has focused more on replication than creation/innovation. However, although diversity is related to innovative and creative products, it also has negative effects on team performance in terms of inefficiencies that arise in team coordination, resulting in poor performance (Ancona and Caldwell 1992, Owens et al. 1998) and to turnover (Jackson et al. 1991).

The potential negative effects of the homogeneity in member skills and experience in naturally occurring OSSD project teams on project innovativeness may however be offset by the positive process gains that result. Members that are familiar with one another may be able to coordinate more effectively and hence perform better (Faraj and Sproull 2000a, Guzzo and Dickson 1996). Strong ties may also facilitate the transfer of the complex knowledge required for effective coordination in software development (Hansen 1999). In addition, the potential dampening impact of the initial project team homogeneity on product innovativeness may be less problematic due to the unique context of open source software development. Open source software development communities generally exhibit a core-periphery structure (Crowston and Howison 2005). Studies have shown that it is the initial project members who are most likely to become the core of the project. Over time as the project grows, the user developer community will also grow, bringing in a wider variety of software use issues in different user contexts that the software will need to address. We argue that the weak ties that these peripheral participants represent will provide the impetus for innovation within the project. Hansen (1999) argues that both weak and strong ties are important. Weak ties are able to search in a broader field for relevant knowledge. So what would be the optimal mixture of strong vs. weak ties or core vs. periphery, or the appropriate timing of changes of such mixtures in OSSD that would foster innovation while at the same time ensuring process efficiencies? We call for additional research in this exciting direction.

To this point our discussion has addressed the question of the implications of homogeneous teams resulting from the team formation process that we observed. But, how homogeneous are the OSSD project teams? Our data only tracks prior project experience. Further research is needed to study the actual variations in group composition of OSSD teams in terms of attitudes, ideological beliefs regarding free and open source software, gender and occupational background among others.

We end our discussion by noting some limitations and additional directions for future research. One limitation may be that we only examine joining behavior within the first two months after project registration. The joining behavior may differ during different stages of project development. While controlling for development stage would shed more theoretical insights, practically many newly registered projects do not define their development stages explicitly, which limits our ability to incorporate this factor into the analysis. An important extension of this paper would be to study the effect of developer joining behavior on the network structural characteristics within project team as well as in the OSS developer network. Furthermore, our data only includes information available from SourceForge.net. Even though SourceForge.net is currently the largest repository, constraining the data collection to one site only may introduce measurement error. For example, even if a developer may have had extensive OSSD experience outside of SourceForge.net, if she joins her first project hosted herein, she would be considered inexperienced.

REFERENCES

Allison, P. D. 1999. *Logistic regression using the SAS system: Theory and application*. SAS Institute, Cary, N.C.

Ancona, D. G. and D. F. Caldwell. 1992. Bridging the boundary: External activity and performance in organizational teams. *Administrative Science Quarterly* **37**(4) 634-665.

Barabasi, A.-L. and R. Albert. 1999. Emergence of scaling in random networks. *Science* **286**(5439) 509-512.

Beal, D. J., R. R. Cohen, M. J. Burke and C. L. McLendon. 2003. Cohesion and performance in groups: A meta-analytic clarification of construct relations. *Journal of Applied Psychology* **88**(6) 989-1004.

Boehm, B. W. 1987. Improving software productivity. *IEEE Computer* **20**(9) 43-57.

Borgatti, S. P. and P. C. Foster. 2003. The network paradigm in organizational research: A review and typology. *Journal of Management* **29**(6) 991-1013.

Burt, R. 1992. *Structural holes: The social structure of competition*. Harvard University Press, Cambridge, MA.

Connolly, T. and B. K. Thorn. 1990. Discretionary databases: Theory, data, and implications. *Organizations and communication technology*. J. Fulk and C. Steinfield, eds. Sage, Newbury, CA, 219-233.

Cook, K. S. and J. M. Whitmeyer. 1992. Two approaches to social structure: Exchange theory and network analysis. *Annual Review of Sociology* **18** 109-127.

Crowston, K., H. Annabi and J. Howison. 2003. Defining open source software project success. *Proceedings of Twenty-Fourth International Conference on Information Systems*, Seattle WA, 327-340.

Crowston, K. and J. Howison. 2005. The social structure of free and open source software development. *First Monday* **10**(2).

Curtis, B., H. Krasner and N. Iscoe. 1988. A field study of the software design process for large systems. *Communications of the ACM* **31**(11) 1268-1287.

Ducheneaut, N. 2005. Socialization in an open source software community: A socio-technical analysis. *Computer Supported Cooperative Work* **14**(4) 323-368.

Faraj, S. and L. Sproull. 2000a. Coordinating expertise in software development teams. *Management Science* **46**(12) 1554-1568.

Faraj, S. and L. S. Sproull. 2000b. Coordinating expertise in software development teams. *Management Science* **46**(12) 1554-1568.

Granovetter, M. 1973. The strength of weak ties. *American Journal of Sociology* **78**(6) 1360-1380.

Gruenfeld, D. H., E. A. Mannix, K. Y. Williams and M. A. Neale. 1996. Group composition and decision making : How member familiarity and information distribution affect process and performance. *Organizational Behavior and Human Decision Processes* **67**(1) 1-15.

Gulati, R. 1995. Social structure and alliance formation pattern: A longitudinal analysis. *Administrative Science Quarterly* **40**(4) 619-652.

Guzzo, R. A. and M. W. Dickson. 1996. Teams in organizations: Recent research on performance and effectiveness. *Annual Review of Psychology* **47** 307-338.

Hansen, M. T. 1999. The search-transfer problem: The role of weak ties in sharing knowledge across organization subunits. *Administrative Science Quarterly* **44**(1) 82-111.

Hars, A. and S. Qu. 2002. Working for free? Motivations for participating in open-source projects. *International Journal of Electronic Commerce* **6**(3) 25-39.

Hertel, G., S. Niedner and S. Herrmann. 2003. Motivation of software developers in open source projects: An Internet-based survey of contributors to the Linux kernel. *Research Policy* **32**(7) 1159-1177.

Hinds, P. J., K. M. Carley, D. Krackhardt and D. Wholey. 2000. Choosing work group members: Balancing similarity, competence, and familiarity. *Organizational Behavior and Human Decision Processes* **81**(2) 226-251.

Hogg, M. A. and D. Abrams. 1988. *Social identifications: A social psychology of intergroup relations and group processes*. Routledge, London.

Howison, J. and K. Crowston. 2004. The perils and pitfalls of mining sourceforge*. Proceedings of the 26th International Conference on Software Engineering (ICSE 2004), Mining Software Repositories Workshop*, Edinburgh, Scotland.

Huang, S. and G. DeSanctis. 2005. Mobilizing informational social capital in cyber space: Online social network structural properties and knowledge sharing*. Proceedings of the 26th International Conference on Information Systems*, Las Vegas, NV, 207-219.

Jackson, S. E., J. F. Brett, V. I. Sessa, D. M. Cooper, J. A. Julin and K. Peyronnin. 1991. Some differences make a difference: Individual dissimilarity and group heterogeneity as correlates of recruitment, promotions, and turnover. *Journal of Applied Psychology* **76**(5) 675-689.

Karau, S. J. and K. D. Williams. 2000. Understanding individual motivation in groups: The collective effort model. *Groups at work: Theory and research*. M. E. Turner, ed. Lawrence Erlbaum Associates, Mahwah, NJ, 113-141.

King, G. and L. Zeng. 2001. Logistic regression in rare events data. *Political Analysis* **9**(2) 137-163.

Lakhani, K. R. and R. Wolf. 2005. Why hackers do what they do: Understanding motivation and effort in free/open source software projects. *Perspectives on free and open source software*. J. Feller and B. Fitzgerald and S. Hissam and K. R. Lakhani, eds. MIT Press, Cambridge, MA, 3-22.

Lerner, J. and J. Tirole. 2001. The open source movement: Key research questions. *European Economic Review* **45**(4-6) 819-826.

Lerner, J. and J. Tirole. 2002. Some simple economics of open source. *Journal of Industrial Economics* **50**(2) 197-234.

Levine, J. M. and R. L. Moreland. 1990. Progress in small group research. *Annual Review of Psychology* **41** 585-634.

Lopez-Fernandez, L., G. Robles and J. M. Gonzalez-Barahona. 2004. Applying social network analysis to the information in cvs repositories. *Proceedings of the 26th International Conference on Software Engineering (ICSE 2004), Mining Software Repositories Workshop*, Edinburgh, Scotland.

Madey, G., V. Freeh and R. Tynan. 2002. The open source software development phenomenon: An analysis based on social network theory. *Proceedings of the 8th Americas Conference on Information Systems*, Dallas, Texas, USA, 1806-1813.

Manski, C. F. and S. R. Lerman. 1977. The estimation of choice probabilities from choice based samples. *Econometrica* **45**(8) 1977-1988.

Markus, M. L. 1987. Toward a critical mass theory of interactive media: Universal access, interdependence and diffusion. *Communication Research* **14**(5) 491-511.

McClelland, D., J. Atkinson, R. Clark and A. Lowell. 1953. *The achievement motive*. Appleton-Century-Crofts, New York, NY.

Moon, J. Y. and L. S. Sproull. 2002. Essense of distributed work: The case of the Linux kernel. *Distributed work*. P. J. Hinds and S. B. Kiesler, eds. MIT Press, Boston, MA, 381-404.

Moreland, R. L. 1999. Transactive memory: Learning who knows what in work groups and organizations. *Shared cognition in organizations: The management of knowledge*. L. I. Thompson and J. M. Levine and D. M. Messick, eds. Lawrence Erlbaum Associates, Mahwah, NJ, 3-31.

O'Reilly, T. 1999. Lessons from open-source software development. *Communications of the ACM* **42**(4) 33-37.

Owens, D. A., E. A. Mannix and M. A. Neale. 1998. Strategic formation of groups: Issues in task performance and team member selection. *Research on managing groups and teams: Composition*. D. H. Gruenfeld, ed., (Vol. 1). JAI Press, Stamford, CT, 149-165.

Powell, W. W., K. W. Koput and L. Smith-Doerr. 1996. Interorganizational collaboration and the locus of innovation: Networks of learning in biotechnology. *Administrative Science Quarterly* **41**(1) 116-145.

Raymond, E. S. 2001. *The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary*. O'Reilly and Associates, Sebastapol, CA.

Roberts, J. A., I.-H. Hann and S. A. Slaughter. 2006. Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the apache projects. *Management Science* **52**(7) 984-999.

Robey, D. and M. Newman. 1996. Sequential patterns in information systems development: An application of a social process model. *ACM Transactions on Information Systems* **14**(1) 30-63.

Ruef, M., H. E. Aldrich and N. M. Carter. 2003. The structure of founding teams: Homophily, strong ties, and isolation among u.S. Entrepreneurs. *American Sociological Review* **68**(2) 195-222.

Sawyer, S., J. Farber and R. Spillers. 1997. Supporting the social processes of software development. *Information Technology & People* **10**(1) 46-62.

Sawyer, S. and P. J. Guinan. 1998. Software development: Processes and performance. *IBM Systems Journal* **37**(4) 552-569.

Schachter, S. 1959. *The psychology of affiliation*. Stanford University Press, Stanford, CA.

Singh, J. 2005. Collaborative networks as determinants of knowledge diffusion patterns. *Management Science* **51**(5) 756-770.

Stewart, D. 2005. Social status in an open-source community. *American Sociological Review* **70**(5) 823-842.

Stewart, K. J., A. P. Ammeter and L. M. Maruping. 2006. Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects. *Information Systems Research* **17**(2) 126-144.

Thye, S. R. 2000. A status value theory of power in exchange relations. *American Sociological Review* **65**(3) 407-432.

Tsai, W. 2001. Knowledge transfer in intraorganizational networks: Effects of network position and absorptive capacity on business unit innovation and performance. *Academy of Management Journal* **44**(5) 996-1004.

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

von Krogh, G., S. Spaeth and K. R. Lakhani. 2003. Community, joining, and specialization in open source software innovation: A case study. *Research Policy* **32** 1217-1241.

Walker, J., S. Wasserman and B. Wellman. 1994. Statistical models for social support networks. *Advances in social network analysis*. S. Wasserman and J. Galaskiewicz, eds. Sage, Thousand Oaks, CA., 53-78.

Wasserman, S. and J. Galaskiewicz. 1994. *Advances in social network analysis*. Sage, Thousand Oaks, CA.

Weiss, M., G. Moroiu and P. Zhao. 2006. Evolution of open source communities. *Open source systems*. E. Damiani and B. Fitzgerald and W. Scacchi and M. Scotto and G. Succi, eds., (Vol. 203). Springer, Boston, 21-32.

Wellman, B. and S. D. Berkowitz. 1998. *Social structures: A network approach*. Cambridge University Press, Cambridge, UK.

Xu, J., Y. Gao, S. Christley and G. madey. 2005. A topological analysis of the open source software development community. *Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS '05)*, Hawaii, HI.

Zander, A. and A. Havelin. 1960. Social comparison and interpersonal attraction. *Human Relations* **13**(1) 21-32.

TABLES

Table 1. Composite Measures of Strength of Prior Collaborative Ties with Project Initiator

Variable	Factor 1 *TieStrengthProduct*	Factor 2 *TieStrengthProcess*
AverageDownloads	0.539	-0.232
AverageDuration	0.732	0.070
AverageHasRelease	0.703	-0.003
AverageDevelopment Status	0.376	-0.177
AverageBytes	0.283	0.053
BothAdminRole	-0.160	0.651
NumCollaborations	0.133	0.795

Table 2. Summary of Measures (See next page – landscape page layout)

Table 3. Descriptive Statistics (Project Level)

	Descriptive Statistics			
Variable	Mean	St. Dev	Min	Max
DeveloperJoin	0.23	0.424	0.00	1.00
InitiatorTieAmount	0.44	0.930	0.00	6.01
InitiatorExperience_t	4.76	1.962	0.40	7.68
InitiatorExperience_p	0.45	0.606	0.00	3.04
DescDetail	5.00	0.600	2.30	5.92
TroveDefined	0.53	0.500	0.00	1.00
AcceptDonation	0.05	0.218	0.00	1.00
TopicPopularity	0.12	0.074	0.00	0.62
Duration	3.96	0.169	3.62	4.22

Table 4. Correlations (Project Level)

Variable	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(1) DeveloperJoin								
(2) InitiatorTieAmount	0.141***							
(3) InitiatorExperience_t	-0.004	0.423***						
(4) InitiatorExperience_p	0.026	0.642***	0.577***					
(5) DescDetail	0.018	0.009	0.025	0.018				
(6) TroveDefined	0.043*	0.025	0.130***	0.079***	0.091***			
(7) AcceptDonation	-0.005	-0.018	0.050**	0.009	0.053**	0.135***		
(8) TopicPopularity	-0.023	0.032	0.108***	0.067***	0.093***	0.224***	0.055**	
(9) Duration	0.018	-0.007	0.144***	0.001	0.026	0.137***	0.051**	0.027
Significance levels: *** 0.01, ** 0.05, * 0.1								

Table 2. Summary of Measures

Variable	Operational Definition
Project Level	
DeveloperJoin (DV)	Binary variable, which equals 1 if at least one developer joined the project within the first one/two months of project initiation (i.e., before Nov 21, 2005), 0 otherwise.
NumJoiningDev (DV)	The number of developers who joined the project within the first one/two months of project initiation (i.e., before Nov 21, 2005).
InitiatorTieAmount	The amount of collaborative ties that the project initiator has prior to project inception, It is calculated as the natural log of the number of distinct developers (+1) who have collaborated with the project initiator on OSS projects at Sourceforge.net.
InitiatorExperience_t	The project initiator's experience in terms of the natural log of the number of days since he/she registered on Sourceforge.net.
InitiatorExperience_p	The project initiator's experience in terms of the natural log of the number of prior projects (+1) that he/she has participated in on Sourceforge.net.
DescDetail	The level of details in project description as measured by the natural log of the number of characters in the project description.
TroveDefined	Indicator variable that is 1 if the project has been defined in the software map of Sourceforge.net, 0 otherwise.
AcceptDonation	Indicator variable that is 1 if the project has been set up to accept donations from users, 0 otherwise.
TopicPopularity	Popularity of the project's topic as measured by proportion of developers working on the top 1000 projects within the topic category.
Duration	Duration of the project's life (log of days)
Developer-Project Dyad Level	
Join (DV)	Binary variable, which equals 1 if the developer joined the project within the first one/two months of project initiation, 0 otherwise.
HasTie	Indicator variable that is equal to 1 if the developer has past collaborative tie with the project initiator, 0 otherwise.
TieStrengthProduct	The strength of a collaborative tie approximated by the product of the collaboration. See Table 1.
TieStrengthProcess	The strength of a collaborative tie approximated by the process of the collaboration. See Table 1.
InitiatorTieAmount	Same as project level.
InitiatorExperience_t	Same as project level
InitiatorExperience_p	Same as project level
DeveloperExperience_t	The developer's experience in terms of the natural log of the number of days since he/she registered on Sourceforge.net.
DeveloperExperience_p	The developer's experience in terms of the natural log of the number of prior projects (+1) that he/she has participated in on Sourceforge.net.
MatchTopic	Indicator variable that is 1 if the developer's prior OSS projects' topics match the project's topic, 0 otherwise.
MatchProgLang	Indicator variable that is 1 if the developer's prior OSS projects' programming languages match the project's programming language, 0 otherwise.
MatchOS	Indicator variable that is 1 if the developer's prior OSS projects' operating systems match the project's operating system, 0 otherwise.
Duration	Same as project level
DescDetail	Same as project level
AcceptDonation	Same as project level.
TopicPopularity	Same as project level

Table 5. Project-Level Regression Results

Variable	Model 1: Logistic Regression Parameter Estimate	Model 1: Logistic Regression Odds Ratio	Model 2: Negative Binomial Regression Parameter Estimate
Constant	-2.629*		-4.285***
InitatorTieAmount	0.516***	1.676	0.848***
InitiatorExperience_t	-0.069*	0.933	-0.134***
InitiatorExperience_p	-0.345**	0.708	-0.500***
DescDetail	0.071	1.074	0.004
TroveDefined	0.266**	1.305	0.262**
AcceptDonation	-0.057	0.945	-0.307
TopicPopularity	-1.078	0.340	-1.016
Duration	0.328	1.388	
Dispersion			4.008
Model Statistics			
Sample Size (*N*)	1780		1780
Likelihood Ratio (χ^2)	54.706***		
Deviance			0.579
Significance levels: *** 0.01, ** 0.05, * 0.1			

Table 6. Descriptive Statistics (Developer-Project Dyad Level)

Variable	Mean	St. Dev	Min	Max
Join	0.14	0.351	0.00	1.00
HasTie	0.02	0.142	0.00	1.00
TieStrengthProduct	-0.01[a]	0.164	-2.56	3.32
TieStrengthProcess	0.01[a]	0.226	-1.06	8.57
InitiatorTieAmount	0.56	1.041	0.00	5.64
InitiatorExperience_t	1.96	1.179	0.09	4.12
InitiatorExperience_p	0.52	0.632	0.00	2.77
DeveloperExperience_t	6.48	1.163	1.25	7.69
DeveloperExperience_p	0.83	0.337	0.00	3.09
MatchTopic	0.14	0.351	0.00	1.00
MatchProgLang	0.15	0.354	0.00	1.00
MatchOS	0.19	0.395	0.00	1.00
Duration	3.97	0.164	3.62	4.22
DescDetail	5.05	0.547	2.94	5.92
AcceptDonation	0.08	0.273	0.00	1.00
TopicPopularity	0.13	0.083	0.00	0.62

Notes: [a] For those developer-project dyads without ties *TieStrengthProduct* and *TieStrengthProcess* are coded as 0.

Table 7. Correlations (Developer-Project Dyad Level)

Variable	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(1) Join								
(2) HasTie	0.159***							
(3) TieStrengthProduct	-0.124***	-0.574***						
(4) TieStrengthProcess	0.122***	0.417***	-0.070***					
(5) InitiatorTieAmount	0.189***	0.184***	-0.079***	0.061***				
(6) InitiatorExperience_t	0.010***	0.117***	-0.055***	0.047***	0.387***			
(7) InitiatorExperience_p	0.055***	0.142***	-0.075***	0.063***	0.652***	0.537***		
(8) DeveloperExperience_t	-0.602***	-0.002	0.041***	0.001	-0.135***	-0.000	-0.038	
(9) DeveloperExperience_p	-0.131***	0.141***	-0.059***	0.091***	-0.052	0.003	0.001	0.352***
(10) MatchTopic	-0.015***	0.198***	-0.100***	0.106***	0.020	0.029***	0.019	0.117***
(11) MatchProgLang	-0.005***	0.206***	-0.097***	0.107***	0.025***	0.040***	0.024*	0.106***
(12) MatchOS	-0.067***	0.119***	-0.022*	0.072***	0.007	0.021	0.022	0.172***
(13) Duration	-0.032***	-0.010***	0.008	-0.005	-0.137***	-0.047***	-0.096***	0.050
(14) DescDetail	0.007***	0.002	0.000	-0.011*	-0.043***	-0.037***	-0.033	-0.001***
(15) AcceptDonation	-0.038***	-0.005**	-0.008	0.004	-0.049***	0.024***	-0.012	0.025
(16) TopicPopularity	0.002	0.024	0.003	0.032**	0.021	0.082***	0.021	0.006
Variable	**(9)**	**(10)**	**(11)**	**(12)**	**(13)**	**(14)**	**(15)**	**(16)**
(1) Join								
(2) HasTie								
(3) TieStrengthProduct								
(4) TieStrengthProcess								
(5) InitiatorTieAmount								
(6) InitiatorExperience_t								
(7) InitiatorExperience_p								
(8) DeveloperExperience_t								
(9) DeveloperExperience_p								
(10) MatchTopic	0.256***							
(11) MatchProgLang	0.220***	0.216***						
(12) MatchOS	0.248***	0.212***	0.305***					
(13) Duration	0.039***	0.006***	-0.002***	-0.004***				
(14) DescDetail	0.005	0.037*	0.013	0.019	0.047*			
(15) AcceptDonation	0.018*	0.013	0.002	0.012	0.030*	0.071*		
(16) TopicPopularity	0.007	0.186***	0.019	0.054**	-0.011**	0.121***	0.018**	

Significance levels: *** 0.01, ** 0.05, * 0.1

Table 8. Logistic Regression Results (Developer-Project Dyad Level Analysis)

Variable	Model 4 Parameter Estimate	Model 4 Odds Ratio	Model 5 Parameter Estimate	Model 5 Odds Ratio	Model 6 Parameter Estimate	Model 6 Odds Ratio
Constant	-2.521		6.509		-6.515*	
HasTie	7.244***	1399.211	5.922	373.031	7.545***	1890.40
HasTie×TieStrengthProduct	-0.122	0.885	-1.135	0.321	-0.491	0.612
HasTie×TieStrengthProcess	1.370***	3.934	2.366	10.653	0.880**	2.411
InitiatorTieAmount	0.124***	1.131	0.162***	1.176	0.141***	1.151
InitiatorExperience_t	-0.092***	0.912	-0.123	0.884	0.022	1.022
InitiatorExperience_p	-0.462**	0.630	-0.528	0.590	-0.978***	0.376
DeveloperExperience_t	-1.611***	0.200	-2.537***	0.079	-1.416***	0.242
DeveloperExperience_p	-1.156***	0.315			0.458	1.580
MatchTopic	0.360	1.434	-2.134***	0.118	0.969***	2.636
MatchProgLang	0.224	1.251	-0.885	0.413	0.794***	2.212
MatchOS	0.039	1.039	-1.466***	0.231	0.375	1.455
Duration	0.105	1.111	-0.881	0.414	0.571	1.770
DescDetail	0.092	1.096	0.005	1.005	0.292	1.339
AcceptDonation	-0.435	0.647	-0.411	0.663	-0.497	0.608
TopicPopularity	-0.297	0.743	0.121	1.129	-0.571	0.565
Significance levels: *** 0.01, ** 0.05, * 0.1						

Table 9. Project Performance Regression Results

Dependent Variable / Regression Framework	*HasRelease* Logistic		*ActivityScore* Linear	*DevelopmentStatus* Cumulative Logit	
Variables	**Parameter Estimate**	**Odds Ratio**	**Parameter Estimate**	**Parameter Estimate**	**Odds Ratio**
Constant	-3.667		50.154		
NumJoiningDev	-0.001	0.999	0.381***	0.053*	1.054
InitatorTieAmount	-0.131*	0.877	0.091	0.123	1.131
InitiatorExperience_t	-0.003	0.997	0.334**	0.024	1.025
InitiatorExperience_p	0.276**	1.318	2.083***	-0.268**	0.765
DescDetail	0.291***	1.337	1.160**	-0.277***	0.758
TroveDefined	1.127***	3.086	4.388***	0.438***	1.550
AcceptDonation	0.515**	1.674	1.443	-0.107	0.898
TopicPopularity	-0.807	0.446	-2.657	0.093	1.097
Duration	0.250	1.284	-9.443	-2.497	0.082
Intercept (Planning)				13.238	
Intercept (Pre-Alpha)				14.094*	
Intercept (Alpha)				14.977*	
Intercept (Beta)				16.251*	
Intercept (Stable)				19.218**	
Model Statistics					
Sample Size (N)	1775		1775	1023[a]	
Likelihood Ratio (χ^2)	165.554***			26.218***	
F Statistic			24.990***		
Adj. R^2			0.109		
Significance levels: *** 0.01, ** 0.05, * 0.1					
Notes: [a] The sample size is 1023 because 752 sample projects had not defined their development status by the time of our collection of project performance data.					