December 2004

The Influence of Network Governance Factors on Success in Open Source Software Development Projects

Glen Sagers
Florida State University

The Influence of Network Governance Factors on Success in Open Source Software Development Projects

Glen W. Sagers
College of Business
Florida State University
Tallahassee, FL U.S.A.
gws02@fsu.edu

Abstract

The phenomenon of open source software has lately caught the attention of both the popular press and many researchers in such diverse fields as MIS, computer science, sociology, and management. The organizational challenges faced by open source development projects are significant, because a project must deal with the complexity of coordinating the efforts of a geographically distributed base of volunteers to create a working software product. Based on a theoretical framework of network governance, the influence of social mechanisms on the coordination and safeguarding of exchanges among project members, and how these influence project success are examined. Using survey and objective data, 38 open source projects are empirically investigated to test a formal path model of network governance in open source projects. The model of network governance provides a good explanation of how governance mechanisms can influence success in open source projects, namely, restricted access to the development team improves coordination within the project and safeguards exchanges among project members. Further, collective sanctions safeguard exchanges among project members, and the importance of reputation to project members aids in managing conflicts within the project. Finally, better coordination allows a project to be more successful, while safeguarding exchanges does not appear to impact project success.

Keywords: Open source, free software, network governance, software development, transaction cost economics, exchange mechanisms

Introduction

Open source software development, consisting of a network of individuals bound together by informal social structures rather than contractual obligations, seems to be a recipe for disaster. The picture of complex software being hacked over by many hands presented by Raymond (2000), and that of hundreds of individuals "contributing to a comparatively small product" (Mockus et al. 2002, p. 320) are difficult to reconcile with the ability to produce anything useful. Yet, as evidenced by a number of high profile open source projects such as Apache, which powers two-thirds of all Web sites (Netcraft.net 2004), the Linux operating system kernel, which is receiving increasing attention from such firms as Oracle (Hayes and Greenemeier 2003), Novell, IBM, and HP (Mims 2004) as a platform worthy of their products, and software which runs the backbone of the Internet, such as Sendmail, and the Bind DNS server, it is certainly possible for networks of volunteers to produce useful software.

Coordination of efforts in software development, as in any other complex task, has long been difficult. Several fundamental factors make coordination of software development challenging (Kraut and Streeter 1995). These include scale of the project, uncertainty, and interdependence. As project size and complexity increase, the direction of individuals' efforts to produce a coherent, functioning, and high quality project becomes more difficult. The efforts of many must be coordinated "so that it [the project] gets done and fits together, so that it isn't done redundantly, and so that components of the work are handed off expeditiously" (Kraut and Streeter 1995, p. 69). Additionally, simply keeping track of who knows how to do something and successfully bringing that expertise to bear on a given project is difficult, and becomes more so with more individuals to manage (Faraj and Sproull 2000).

In general, however, OSS projects do not depend on the same formal mechanisms as their proprietary counterparts, such as contracts and organizational hierarchies to coordinate their efforts, but rather rely on informal relationships between developers. These informal coordination mechanisms raise a paradox in inducing volunteers to perform tasks. How can project administrators coordinate the work of these volunteers if they cannot order them to perform a task? How can workers be induced to complete tasks if there is no threat of loss of employment or loss of wages? Open source software development seems like a recipe for chaos, with people doing what they want, when they want to do it, and ignoring requests for order. In reality, open source software development works surprisingly well. The answer to this seeming paradox lies in the mechanisms used to sustain informal relationships through a network form of governance. Network governance uses informal social mechanisms to coordinate and safeguard interactions between actors in a network, and provides a theoretical framework for examining how the efforts of a worldwide programming team of volunteers can be coordinated effectively.

If this type of governance does occur, the degree of utilization of network governance should influence the success of a given project. In this light, the following research questions are proposed:

1. *How do informal network forms of organization, such as OSS projects, apply social mechanisms to coordinate work and safeguard exchanges among project members?*

2. *Does better coordination and safeguarding affect project success?*

This paper is organized as follows. First, the concept of network governance is discussed, grounded in transaction cost economics. A brief description of open source software is given, and a formal model of network governance in OSS projects is introduced. This model is then tested empirically using data gathered from a survey of participants (developers and users of the software) in a number of open source projects. Finally, the implications of these findings for OSS projects are discussed.

Theories of Exchange or Governance

Transaction cost economics (TCE) provides a framework for comparing various alternative forms of governance that are used to coordinate actions and safeguard exchanges against acts of self-interest (Williamson 1994). In terms of coordination, there are costs associated with transactions and the effective integration of actions. Regardless of how a group of individuals or firms is organized, if they engage in transactions with each other, costs will be incurred. Different types of organizations have differing transaction costs, however, due to the utilization of different coordination and governance mechanisms.

Originally, TCE was used to explain the existence of the firm, suggesting that firms emerged in response to high transaction costs in the marketplace, organizing production hierarchically, rather than undertaking market-based transactions. Under TCE, exchanges that take place in a marketplace are governed by individual contracts for each exchange transaction. These contracts must be negotiated separately, and costs are incurred with each contract (Coase 1993). Alternatives to market forms of exchange emerge to reduce these costs. The forms of governance that emerge are varied, depending on the conditions under which they transpire, however, all have the minimization of transaction costs as a goal (Williamson and Winter 1993). Firms are one type of organization that have been characterized by Coase (1993) as emerging when the costs of organizing production internally are less than the costs of organizing production utilizing market price mechanisms. However, firms still incur costs based on the fact that there are costs to the hierarchical management structure (Williamson and Winter 1993). In summary, then, one purpose of effective governance structures is to reduce coordination costs.

In addition to coordination, governance structures also serve to safeguard exchanges. TCE is grounded in the behavioral assumption that individuals and firms are opportunistic, meaning that they are given to maximizing self-interest and are willing to transact with guile to achieve their goals. Commitments to behave in a responsible manner that are not backed by contract or other credible collateral are assumed unlikely to be carried out and unenforceable under TCE (Williamson and Winter 1993, p. 92). Thus, TCE proposes that organizations exist because they have the ability to exercise controls on opportunism that are not available in the marketplace by imposing a hierarchical governance structure (Ghoshal and Moran 1996).

While this view of opportunism in human nature has been attacked on many grounds, as summarized by Ghosal and Moran (1996), without opportunism, "most forms of complex contracting and hierarchy vanish" (Williamson 1993, p. 97). Additionally, Williamson (1993) suggests that governance structures mitigate opportunism, and that sanctions are required to guarantee that those who voluntarily comply with contracts are not taken advantage of by those who do not. Taken together, these statements indicate that some form of governance is necessary to coordinate actions within our human, imperfect, organizations, and to force compliance through organizational rules, policies, and sanctions. Thus, the second purpose of effective governance structures is to safeguard exchanges by reducing opportunistic behavior.

This returns us to the initial paradox. Without contracts to govern the actions of the volunteer labor force of open source development projects, how can project leaders and managers coordinate actions? How can they guard against acts of opportunism that might undermine project success? The success of OSS development projects suggest that other forms of governance may be effective in coordinating actions and safeguarding exchanges. While firms were the first alternative to markets, other hybrid types of organizations, such as franchises and joint ventures, also emerged to combat some of the costs of management and bureaucracy associated with firms (Williamson and Winter 1993). Recently, network governance has received attention as an additional alternative form of governance under certain circumstances.

Network governance is a substitute for the more classical market and hierarchical governance forms. This type of governance has been called by a number of names, such as network organization, network forms of organizations, interfirm networks, and organization networks, all of which refer to styles of coordination between entities (i.e., firms, business units, or individuals). This form of governance is "characterized by informal social systems rather than bureaucratic structures within firms, and formal contractual relationships between them" (Jones et al. 1997, p. 911). Many characteristics found in market or hierarchical organizations are not present in network governance. Networks rely on loose ties between entities in the network to govern transactions that would require formal contracts in other types of governance. Network governance involves a "select, persistent, and structured set of autonomous [entities] engaged in creating products or services based on implicit and open-ended contracts" where the contracts are socially rather than legally binding (Jones et al. 1997, p. 914). This loose structure allows networked entities to adapt to environmental changes and coordinate actions, as well as safeguard exchanges against the hazards of opportunism.

A number of characteristics of network governance are found in OSS projects. OSS projects involve coordinating the efforts of a group of programmers, typically volunteers, who are engaged in creating a software product with no formal contracts or commitments to the project or its members. These members persist in the project due to social standards in the OSS community, a desire to increase their own reputation, a love of the project and writing code, and the open source philosophy, rather than any legal obligation to do so (Lakhani et al. 2002). Additionally, OSS projects use informal social mechanisms to control the actions of project members to safeguard exchanges, such as sanctioning improper behavior, restricting access to the development team, and utilizing the importance of reputation of individuals within the project (Markus et al. 2000).

In summary, OSS projects can be conceptualized as utilizing network governance and informal social mechanisms to coordinate actions and safeguard exchanges.. The essential characteristics of network governance—informal social structures, (somewhat) persistent membership (Markus et al. 2000), lack of contractual obligations, and participation in creating a software product—are present in open source projects. This study, therefore, proposes to test a formal model of network governance in OSS projects to determine whether the utilization of informal social mechanisms to improve coordination and safeguarding leads to success in open source projects.

Model and Hypotheses

The formal model of network governance proposed by Jones et al. (1997) was adapted for this study. To summarize, the adapted model proposes that restricted access to the development team of an OSS project will lead to better coordination of efforts and better safeguarding of exchanges among members of the project. Likewise, the utilization of collective sanctions and the importance of reputation to project members will result in better safeguarding of exchanges. Improved coordination and better safeguarding of exchanges will lead to success in the project. The theoretical model is summarized in Figure 1.

Restricted access refers to a strategic reduction in the number of exchange partners within a network. This reduction in the number of network participants allows for more frequent exchanges, a necessary precondition for the emergence of network governance. Restricting access to the network also reduces coordination costs by allowing communication protocols and routines to be established (Jones et al. 1997). These communication protocols and routines better enable the coordination of activities with fewer costs and less effort. Restricting access to the network also helps to minimize differences in network members' skills, expectations, and goals.

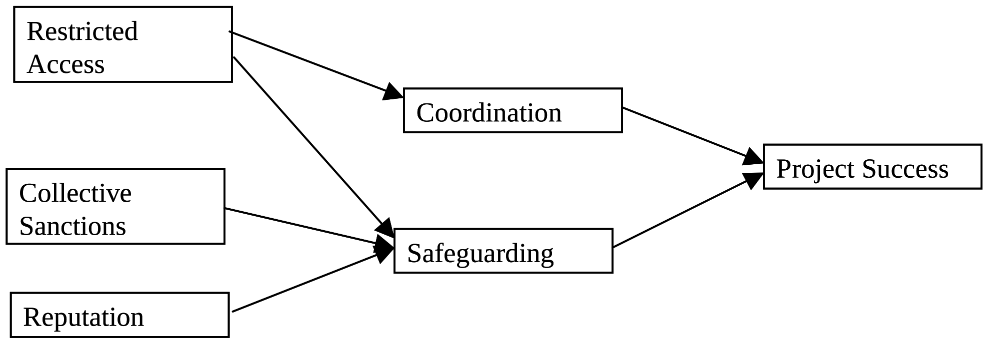

Figure 1. Theoretical Model of Network Governance in Open Source Projects

In OSS projects, open source licenses do not allow restricting access to the software product. Thus, in this context, restricting access to the network occurs by limiting the number of individuals allowed to change the source code. Individuals wishing to become developers must go through a process of vetting so that project administrators can ensure that the individual's skills are adequate for the task. Restricting access to the development team reduces the amount of coordination required when the individual begins coding a section of the project. Additionally, by restricting access to the development team, it can be ascertained whether an individual's goals are congruent with those of the team. When the expectations and goals of the individual are aligned with the expectations and goals of the development team, the amount of coordination needed to set goals is reduced, thus coordination can be improved (Jones et al. 1997). This leads to the following hypothesis:

> *H1: Projects characterized by higher levels of restricted access to the development team will have better coordination.*

Restricting access to the development team safeguards exchanges by increasing identification with collective norms of the project, and by reducing the amount of monitoring required. A smaller number of members on the development team allows individuals to interact more frequently with each other, enhancing the sense of team identity and shared goals. This allows stronger bonds to be formed among members, leading to increased commitment and better identification with the development team and its goals (Granovetter 1973). When commitment is increased, participants in exchanges can be certain that the exchange is safeguarded against deceptive behaviors or other malfeasance—in other words, opportunism in these exchanges is reduced.

Restricting access to the development team safeguards exchanges by decreasing the amount of monitoring that is required and increasing the interaction of individuals within the team (Jones et al. 1997). Fewer members also means that the rules and procedures of the team are more easily enforceable, due to the decreased number of individuals who must be monitored for compliance. By restricting membership to only those individuals who are known to work well and to uphold standards of the project, a project development team can reduce the amount of time spent monitoring. Thus the prediction is that

> *H2: Projects characterized by higher levels of restricted access to the development team will have better safeguarding of exchanges between members.*

A second social mechanism that safeguards exchanges within a network is the use of collective sanctions. Collective sanctions are punishments enforced by the network membership on individual entities who violate network norms, values, or goals, and have the effect of increasing the incentive to follow norms. This safeguards exchanges by increasing the costs of misconduct, decreasing monitoring costs to any one party, and providing incentives to monitor partners (Jones et al. 1997). If an individual entity plans to stay with a network, the mere threat of collective sanctions may be enough; indeed, the social costs of enforcing the sanction may be too high (Olson 1965).

There are a number of types of collective sanctions available to open source projects. These include, but are not limited to, flaming, shunning, and expulsion from the project (Markus et al. 2000). Flaming is an exchange of inflammatory e-mail and has been seen many times on open source mailing lists, including a high-profile example where leaders of the movement threatened each other (Kanhey 1999). Shunning consists of ignoring or ostracizing a member of the project for misbehavior. The most drastic punishment for most project members would be expulsion from the project on a temporary or permanent basis. If it is known by individuals within the project that inappropriate actions may result in these sanctions, the individuals will be more likely to avoid these actions. It is thus proposed that

H3: Projects characterized by higher utilization of collective sanctions will have better safeguarding of exchanges between project members.

A third social mechanism that safeguards exchanges within a network is the importance of reputation to individual network members. In general, exchanges are safeguarded because individuals care about their reputations (Markus et al. 2000). Reputation refers to others' perceptions of one's overall quality or character, and has been shown to play a role in selection of entities within a network (Gemser and Wijnberg 2001). Reputation is important in network governance because it relays information about prior behaviors and serves to deter opportunism and guile (Jones et al. 1997).

In OSS projects, the perceived importance of reputation among project members safeguards exchanges by spreading information about behavior among parties (Jones et al. 1997). Prior research suggests that individual reputation within the project is a major reason why OSS projects are successful (Markus et al. 2000). Further, the reputation of individuals within the larger OSS community has been suggested as a reward for doing work on open source projects (Taylor 1999). Individual reputation has also been shown to be a motivator for individuals to donate their time to open source development (Lakhani et al. 2002). Thus, when reputation is perceived as important, individuals will be motivated to protect their reputations by avoiding deceptive or self-interested behavior.

H4: Projects characterized by higher levels of the perceived importance of reputation among members will have better safeguarding of exchanges between project members.

Increased coordination prevents duplication of work (Kraut and Streeter 1995), allowing better coordination of tasks among network members. This reduces the amount of time spent managing resources, and allows the many modules of a complex software product to work together (Mockus et al. 2002). Better coordination also allows project members to spend more time on writing code, adding new features, fixing bugs, and generally making the software more useful and feature-rich. When project members are able to spend their time in this fashion, the project will be more successful. This leads to the following hypothesis:

H5: Projects characterized by higher levels of coordination of exchanges among project members will be more successful.

Open source project members engage in transactions with each other, and with other projects. These exchanges can take the form of interpersonal communications, ideas about the software, or actual code. These transactions are subject to transaction costs, including opportunism. To guard against opportunism and to facilitate easy exchanges, open source projects utilize rules and procedures and must be able to manage conflict when it occurs. The rules and procedures that govern a project vary between projects (Markus et al. 2000), but all are concerned with making sure that exchanges flow freely. Similarly, the specific manner in which conflict is resolved may vary between projects, and the outcomes may not be acceptable to every individual or project involved, but in order for development to progress, conflict must be managed (Markus et al. 2000; Sawyer 2000).

Rules and procedures govern the operations of the project (Pinto et al. 1993). These rules and procedures may cover when someone is allowed to join the development team or how new leaders of the project are selected (Markus et al. 2000). Rules also exist to govern such day-to-day administrative details as who has read-write access to the source code, or if and when developers will meet in a chat room to discuss development tasks and milestones for the project. Whatever form they take, these rules and procedures safeguard exchanges. Similarly, when the project effectively manages conflict, members may be assured that conflict will not bring the development to a screeching halt. When project members feel that their exchanges are protected from possible malfeasance, they will be more willing to contribute to the project. These contributions, in terms of actual code, ideas, expertise, or willingness to help others, aid in the success of the project. Thus, it is proposed that

H6: Projects characterized by higher levels of safeguarding of exchanges between members will be more successful.

Measures

The survey measures for this study were largely derived from previously validated scales utilized in published studies. In addition to the survey scales, objective data was collected from the SourceForge Website. This additional data helped to mitigate potential mono-method bias. While several of the measures (e.g., coordination and conflict management) may be considered processes and not variables in the traditional sense, the studies from which the scales were taken utilized them as variables.

Restricted access was measured as the ratio of the number of developers listed on the project SourceForge page divided by the number of individuals who have posted to the forums during this six week time period. This allows the capture of actual amount of restricted access, rather than intentions to use restricted access or perceptions of the level of use of restricted access.

Collective sanctions are measured by project members' perceptions of the utilization of flaming, refusal to cooperate, public humiliation, and expulsion from the OSS project. These constructs were suggested by Markus et al. (2000).

The importance of reputation to participants in a project is measured by the project members' perceptions of concern for their own reputation within the project. Items for this scale were adapted from Constant et al. (1996).

Coordination is measured by the ability of project leaders to know where expertise is located within the project. Items for this scale were adapted from Kraut and Streeter (1995) and from Janz et al. (1997).

Safeguarding indicates the extent to which project members feel that their exchanges are safe from deception or other malfeasance, as measured by the ability of the project to manage conflict (Sawyer and Guinan 1998), and by the rules and procedures that are in place to safeguard exchanges (Pinto et al. 1993).

Success is measured in several ways. Objectively, the measures used were the age of the project (since its founding on SourceForge) and the ratio of total bug reports and feature requests to open bug reports. Feature requests were objectively determined. A long-lived project which is still active may be considered successful, as activity among developers and users is indicative of success (Crowston et al. 2003), however project age alone is not a valid measure of success. Likewise, a high ratio of open bug reports and feature requests to total reports and requests suggest developers are not actively fixing bugs or adding new features. This measure was adapted from a measure suggested by Crowston et al. (2003)—that the time required to fix bugs was a measure of success in OSS projects.

Finally, subjective feelings of the project members relating to the value, performance, and utility of the software product were measured. This self-rated perception of software performance scale from Hartwick and Barki (1994) was used based on the concept of determining subjective feelings of project members to the software, as suggested by Crowston et al.

Population and Sampling

There are several sites which host or list a number of open source projects, including SourceForge, Freshmeat, and the GNU projects' Savannah. The largest of these, SourceForge, currently hosts more than 80,000 projects. These projects cover a wide range of software types, including games, multimedia, applications, network and Internet software, operating systems, and server software, and vary in degrees of maturity such as planning, alpha and beta stages, and mature, stable software. The advantage of using a site which hosts multiple projects is that data collection may be standardized, in that the same data is available for all projects. The disadvantage of such a site is that all successful projects are not included in the sample as a number of successful projects have their own Web sites, and thus are not listed on these sites. In order to standardize data collection, and apply consistent criteria to the selection of projects, it was decided to investigate only projects hosted on SourceForge. Although these projects and their members do not represent the totality of open source development, there are enough projects that they may fairly be considered representative of much of the larger body.

The projects were selected from the "Top Forum Posts Count" list maintained by SourceForge (2004a), as these projects, by definition, have active participants who communicate with one another. This communication was necessary as the network structure is generated by communications, interaction, and exchanges among project members. In projects without active discussion, different governance forms would likely emerge. It was further decided to only include projects that had active participation on the "Open Discussion" forums, rather than including the forums for help, bugs, and feature requests, in order to capture general discussions of the software rather than specific issues. Finally, it was decided to accept only projects that did not have outside mailing lists or other outside venues for discussion. This was done to ensure that all discussions between individuals took place on the forums, and thus were visible for analysis.

All 100 projects in this list were examined for a minimum of 10 messages in the "Open Discussion" forums during the time period of January 1, 2004, to February 18, 2004. This allowed enough time to have interactions between developers, while keeping data collection to a manageable level. This filtering left 44 projects. E-mail addresses were extracted from posts in the forums and each participant in the forum discussions was contacted by individual e-mail with an invitation containing a link to the Web-based survey.

Analysis

A total of 1,672 unique e-mail addresses were gathered, and a personal e-mail containing a unique, personalized link to the survey Website was sent to each individual. Of the initial mailing, 122 were returned as undeliverable, for a total of 1,550 potential respondents. There were 15 duplicate individuals across projects; when these were listed as developers in one project, they were assigned to that project, otherwise they were assigned to the first project (alphabetically), in which they appeared. A total of 318 usable responses were received, for a response rate of 21 percent. Six projects had a response rate less than 10 percent. These projects were removed from the analysis, leaving a total of 38 projects for analysis. The average size of each project was 40.8 members, with an average of 8.4 responses per project.

In order to test for response bias, demographic variables collected from respondents were compared with published accounts that surveyed only developers in OSS projects. In this sample, the average age was 37.6 years (range 18 to 73 years), slightly older than the mean of 30 years found by Lakhani et al. (2002). The respondents to this survey were 96 percent male, slightly less than the 98 percent reported by Lakhani et al., or the 99 percent reported in the FLOSS survey (2002). The respondents to this survey lived in the following geographical areas (numbers from Lakhani et al. are in parentheses): Americas 47.9 percent (46.9 percent), Europe 39.3 percent (42.4 percent), Africa, Asia, Pacific 10 percent (10.7 percent). Thus, there does not seem to be any significant response bias compared to findings from other research.

Partial least squares was selected as the tool to analyze the hypothesized model. PLS is a structural equation modeling technique which allows the simultaneous testing of the reliability and validity of measures and constructs, and estimation of the relationships among these constructs (Wold 1982). PLS was selected since it is able to work with small sample sizes. It also handles errors in measurement, including non-normal distributions, and it can work with error in theory such as unexplained variance and surplus meaning in constructs (Barclay et al. 1995). PLS can also analyze structural models with multi-item constructs, including those with indirect and interaction effects, and handles small samples The required sample size is 10 times the number of the predictors from either the indicators of the most complex formative construct, or the largest number of antecedent constructs leading to an endogenous construct, whichever is greater (Wold 1982).

The data from this study was aggregated by project, using the mean of individual responses. In order to be able to aggregate responses to a group level, within-team and between-team variance must be tested by means of intra-class correlations (ICC). This is done to insure that members of a project responded similarly to questions, indicating that a unique group influence exists. If the ICC is zero, individuals within the project are no more alike than individuals in other projects; if the ICC is one, all participants in a project have the same score. Table 1 provides the ICC values for the variables used in this study. The range of ICCs is 0.73 to 0.99, indicating a very high level of agreement within teams.

PLS models are analyzed in two stages: first, the convergent validity of the constructs is tested using the average variance extracted (AVE) values and, second, the discriminant validity of the sample is tested by comparing the AVE with the squares of the correlations among the latent variables (Chin 1998).

For the first step of PLS analysis, the AVE values should be greater than a 0.50, meaning that 50 percent or more of the variance has been accounted for. Additionally, the inter-item reliability of items that make up a theoretical construct must be validated. PLS uses the composite reliability (ICR), a measure similar in purpose to Cronbach's alpha, which was developed by Werts et al. (1973). Acceptable values for ICR should exceed 0.7 (Fornell and Larcker 1981). Results of this analysis are shown in Table 1. All AVE values were above the 0.50 cutoff, and all ICR values were above 0.7, indicating that there was adequate convergent validity in the measurement model.

In the second step of PLS analysis, the AVE values are compared with the square of the correlations between constructs. The AVE value for a given construct should be greater than the correlation between that construct and any other construct. The square roots of the AVEs are presented on the diagonal of Table 1, and all are greater than the off-diagonal elements, indicating discriminant validity.

An additional evaluation method for determining convergent and discriminant validity is to examine the factor loadings of each indicator. Each indicator should load higher on its respective construct than on any other factor (Chin 1998). Factor loadings and cross-loadings for the multi-item measures were calculated from the output provided by PLS. Inspection of the loadings and cross-loadings, presented in Table 2, validates that the indicators have adequate discriminant and convergent validity.

Table 1. Constructs, Intra-Class Correlations, Average Variance Explained, and Correlations of Constructs and AVE Values

	Construct	# of Items	ICC	AVE	ICR	1	2	3	4	5	6	7	8	9
1	Restricted Access	1	0.97	n/a	n/a	**1.00**								
2	Collective Sanctions	4	0.99	0.70	0.91	0.25	**0.84**							
3	Reputation	3	0.83	0.82	0.93	0.33	0.19	**0.90**						
4	Coordination	3	0.72	0.56	0.79	-0.40	0.00	0.07	**0.75**					
5	Conflict Management	3	0.82	0.86	0.95	0.18	0.41	0.47	0.51	**0.93**				
6	Rules and Procedures	2	0.88	0.85	0.92	0.23	-0.47	-0.05	-0.40	-0.38	**0.92**			
7	Self-rated Performance	4	0.95	0.69	0.92	-0.18	-0.06	0.22	0.62	0.49	-0.25	**0.83**		
8	Project Age	1	1.00	n/a	n/a	0.10	0.35	-0.01	0.12	-0.21	-0.12	0.20	**1.00**	
9	Open To Total Issues	1	0.98	n/a	n/a	0.23	0.16	0.38	-0.24	0.00	-0.11	-0.01	0.19	**1.00**

Table 2. Factor Loadings and Cross-Loadings

	Collective Sanctions	Reputation	Team Coordination	Conflict Management	Self-Rated Performance	Rules and Procedures
CS1	0.86	0.02	0.04	0.27	0.04	0.46
CS2	0.86	0.17	0.04	0.29	0.13	0.39
CS3	0.78	0.23	0.25	0.32	0.04	0.39
CS4	0.86	0.26	0.14	0.47	0.07	0.33
REP1	0.20	0.94	0.10	0.50	0.27	0.10
REP2	0.20	0.91	0.07	0.44	0.21	0.01
REP3	0.08	0.86	0.02	0.25	0.07	0.02
TC1	0.11	0.20	0.79	0.55	0.50	0.37
TC2	0.15	0.18	0.74	0.54	0.52	0.66
TC3	0.21	0.17	0.71	0.10	0.39	0.05
CM1	0.55	0.38	0.47	0.90	0.48	0.45
CM2	0.26	0.46	0.49	0.94	0.45	0.30
CM3	0.31	0.47	0.47	0.95	0.42	0.31
PERFWV	0.14	0.23	0.55	0.40	0.82	0.22
PERFUU	0.00	0.14	0.35	0.34	0.79	0.25
PERFGB	0.00	0.16	0.43	0.40	0.84	0.06
PERFTT	0.03	0.19	0.65	0.44	0.86	0.28
RP1	0.34	0.01	0.18	0.24	0.09	0.89
RP2	0.49	0.08	0.51	0.43	0.33	0.95

Hypothesis and Model Testing

The hypothesized relationships in the theoretical model were estimated using the bootstrapping function in PLS Graph 2.91, using 200 iterations. The explanatory power of the structural model is evaluated by the R^2 value, which represents the variance accounted for in the final construct. For each hypothesis, the t-statistics for the standardized path coefficient and calculated p-values based on a two-tail test with a significance level of 0.05 was assessed. The results of the PLS analysis of the measurement model are shown in Figure 2.

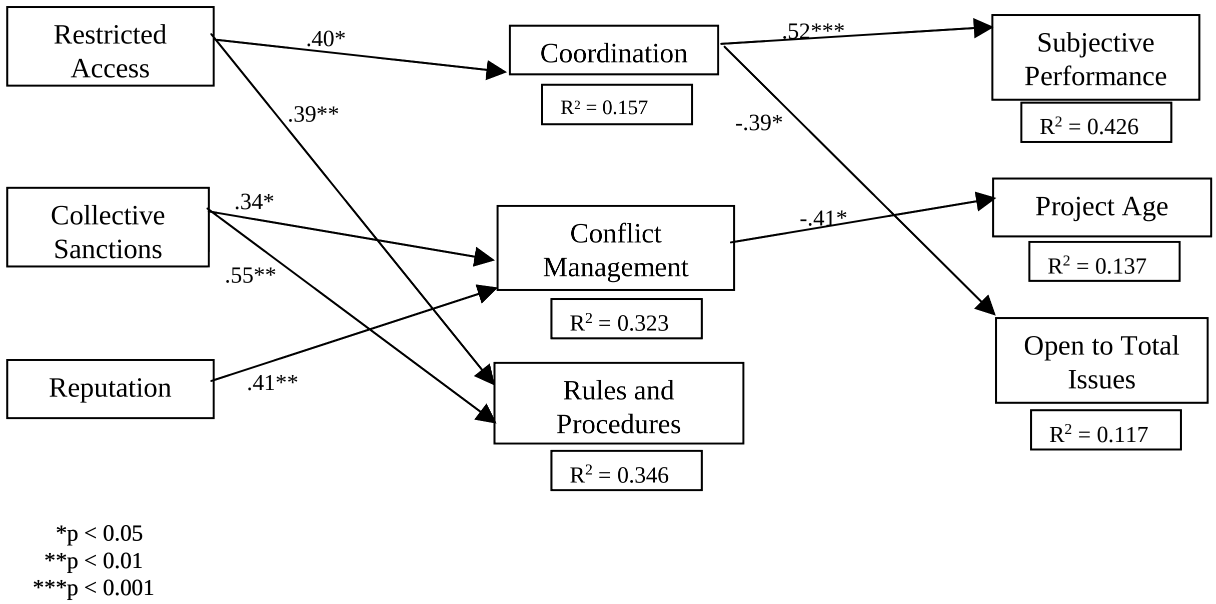

Figure 2. Measurement Model Results

The R^2 value for coordination was 0.16, R^2 for conflict management was 0.32, and R^2 for rules and procedures was 0.35. Hypothesis 1 proposed that restricting access to membership in the development team would improve coordination. The path between restricted access and coordination was significant and positive ($\beta = 0.40$, $p < .05$).

Hypothesis 2 proposed a link between restricted access and safeguarding exchanges between project members. Safeguarding was assessed as consisting of both conflict management and rules and procedures. The path between restricted access and conflict management was not significant ($\beta = 0.04$, n.s.), and the path between restricted access and rules and procedures was positive and significant ($\beta = 0.39$, $p < .01$).

Hypothesis 3 suggested that higher levels of collective sanctions will lead to better safeguarding of exchanges. The path between collective sanctions and conflict management was positive and significant ($\beta = 0.34$, $p < .01$); the path between collective sanctions and rules and procedures was also positive and significant ($\beta = 0.55$, $p < .01$).

Hypothesis 4 predicted that in projects where members view reputation as important, there will be better safeguarding of exchanges. The path between reputation and conflict management was positive and highly significant ($\beta = 0.41$, $p < .01$). The path between reputation and rules and procedures was not significant ($\beta = 0.39$, n.s.).

Hypothesis 5 suggested that better coordination will lead to project success. Hypothesis 6 suggested a link between safeguarding of exchanges and project success. Three different measures of project success—self-rated performance, project age, and open to total software fixes—were assessed.

The R^2 for the self-rated performance model was 0.43. The path between coordination and self-rated performance was significant ($\beta = 0.52$, $p < .001$); the paths between conflict management and self-rated performance as well as rules and procedures and self-rated performance were not significant ($\beta = 0.24$, n.s.; $\beta = -0.06$, n.s.).

The R^2 for the project age model was 0.14. The path between team coordination and project age was not significant ($\beta = 0.25$, n.s.). The path between conflict management and project age was negative and significant ($\beta = -0.41$, $p < 0.05$), while that between rules and procedures and project age was not significant ($\beta = 0.17$, n.s.).

The R^2 for the open to total issues model is 0.12. The path between coordination and the ability to close issues was negative and significant ($\beta = -0.39$, $p < 0.05$).

Discussion

Many published studies have examined one or a small number of open source projects as case studies. This study attempts to advance open source research to model and hypothesis testing;, and represents an attempt to confirm the model proposed by Jones et al. (1997) and the ideas proposed by Markus et al. (2000). The results of this study provide both additional empirical support for the network governance model and a useful framework for studying success in open source projects. Overall, it appears that the network governance model fits at least the selected open source projects quite well. The model provides a useful framework for studying governance and organizational structures in open source projects. The study also confirms earlier predictions (Markus et al. 2000) that restricted access, collective sanctions, and reputation are important to the success of a virtual organization.

The first research question for this study asked, “How do OSS projects apply informal social mechanisms to support the coordination and safeguarding of exchanges among project members?” In terms of coordination, the results indicate that projects may improve their coordination by restricting access to the project development team. This supports the general model of network governance proposed by Jones et al., which suggests that by restricting access to the network, coordination is improved by allowing members of the team interact more frequently, to bring their expectations and goals into congruence, and to develop shared routines for working together. Although this study only looked at the effects of restricted access on coordination, further research should examine how restricted access affects project team interactions.

In order to safeguard exchanges between project members, OSS projects may draw upon collective sanctions and, to some extent, restricted access and reputation. In this study, safeguarding exchanges was measured by both conflict management and the use of rules and procedures. The results indicate that by restricting access to the development team, a project may better develop rules and procedures to govern behavior among members. Restricting access to the development team, on the other hand, does not improve the conflict management portion of safeguarding exchanges between project members.

Collective sanctions safeguard exchanges by both improving conflict management and enabling development of rules and procedures within the project team. Consistent with the theoretical model, collectively imposing sanctions on individuals who violate project norms or deceive other project members allows projects a means of guarding against the opportunism that TCE claims exists in all transactions.

Finally, reputation plays a role in managing conflict within the project; however, reputation does not seem to influence the development of rules and procedures. When members of the project perceive reputation as an important part of their identity, they are more likely to manage conflict within the project in order to preserve their identity. Despite the fact that there are no formal, legally enforceable contractual relationships among project members, network governance mechanisms allow well-coordinated exchanges among members to take place in safety.

The second question answered by this study is, “How do coordination and safeguarding of exchanges affect project success?” By coordinating their efforts, OSS projects improved members’ perceptions of the quality, value, and performance of the software project. These perceptions of the software are important, and are analogous to the construct of user satisfaction that is widely used in MIS literature (Ives et al. 1983; Moore et al. 1991), and to the concept of developer satisfaction with the project proposed by Crowston et al. (2003).

Contrary to expectations, the findings of this study suggest that better coordination of efforts within the project is negatively related to the ability to close bugs or add features. This finding was the most surprising result of the study. From this, it appears that a single individual who is willing and able to fix bugs and add new features to the code is more effective than a group coordinating on the same tasks. If this is the case, it adds a new dimension to one of the claimed benefits of open source: that “with enough eyeballs, all bugs are shallow” (Raymond 2000). While many eyeballs may be useful in finding bugs, it may be sufficient for a single individual to fix them.

Also contrary to expectations, the management of conflict within a project, as a component of safeguarding, was negatively correlated with the success measure of project age. This is the opposite of the relationship proposed by the network governance model. This could imply that younger projects are better able to manage conflict within the project; alternatively, it may imply that older projects have less conflict to manage. The concept that younger projects may be better at managing conflict is in line with the claim of Markus et al., who suggest that conflict within an open source project is not generally detrimental, as long as it is handled well. This would suggest that projects that are better able to manage conflict are more successful, rather than successful projects having less conflict. Given that only one path between safeguarding and success was significant, it appears that safeguarding of exchanges between members of the project may not be strictly necessary for a project’s success.

Limitations and Future Research

While this study supported the ideas that projects which utilize a network form of governance do indeed utilize social mechanisms to coordinate work and that this coordination enhances project success, there are limitations. First, this study examined a small number of projects that, while heterogeneous in terms of type of software and number of members, were all hosted on one site. While this hosting gave uniformity in data collection, it may also influence governance structures by virtue of the tools provided for collaboration and coordination. At least one open source developer has indicated that the tools may shape the interactions and exchanges between project members (Behlendorf, in DiBona et al. 1999, pp. 169-170).

While the sampling methodology limits the ability to generalize outside the SourceForge arena to projects with their own Websites, development servers, and mailing lists, there is some evidence that similar informal social mechanisms are used for coordination and safeguarding in these larger, more mature projects (Markus et al. 2000; Mockus et al. 2002). While specifics of how individuals work together may vary by project, as long as similar informal social mechanisms are in place, it may be reasonably expected that they will be used to increase coordination and safeguard exchanges. Further studies, with a larger number of projects from multiple Websites, could help determine whether tools influence structure, and also whether other as yet unexplored variables may influence project success.

Second, studies utilizing a greater number of success measures in OSS projects may also give a better indication of how the mechanisms of coordination and safeguarding influence project success. In addition, the full model of network governance should be tested to determine how the additional factors in the model affect the outcome. Finally, the negative pathways found in this model should be investigated, particularly that path leading from coordination to the ability to fix bugs and add new features. This should be investigated in terms of separating bug fixing from feature requests, to determine whether bug fixing is different from adding features.

Conclusion

In summary, this study found that the network governance model, as proposed by Jones et al. (1977), provides a good description of the forms of governance used in open source projects. The key findings of this study are that restricted access plays an important role in coordinating among project members; that collective sanctions work to safeguard exchanges among members of the project; and that improving coordination among project members increases project success. This study also provides a basis for future research on how open source projects are governed. Having a tested model of governance will allow the empirical exploration of different facets of success in open source development with a solid theoretical core.

References

Barclay, D., Higgins, C., and Thompson, R. "The Partial Least Squares (PLS) Approach to Causal Modeling: Personal Computer Adoption and Use as an Illustration," *Technology Studies* (2:2), 1995, pp. 285-309.

Chin, W. W. "The Partial Least Squares Approach to Structural Equation Modeling," in *Modern Methods for Business Research*, G. A. Marcoulides (Ed.), Lawrence Erlbaum, Mahway, NJ, 1998, pp. 295-336.

Coase, R. "The Nature of the Firm," in *The Nature of the Firm*, O. E. Williamson and S. G. Winter (Eds.), Oxford University Press, Oxford, 1993, pp. 18-33.

Constant, D., Sproull, L., and Kiesler, S. "The Kindness of Strangers: The Usefulness of Electronic Weak Ties for Technical Advice," *Organization Science* (7:2), 1996, pp. 119-135.

Crowston, K., Annabi, H., and Howison, J. "Defining Open Source Project Success," in *Proceedings of the 25th International Conference on Information Systems*, S. March, A. Massey, and J. I. DeGross, Seattle, WA, 2003, pp. 327-340.

DiBona, C., Ockman, S., and Stone, M. *Open Sources Voices from the Open Source Revolution* (1st ed.), O'Reilly, Beijing, 1999.

Faraj, S., and Sproull, L. "Coordinating Expertise in Software Development Teams," *Management Science* (46:12), 2000, pp. 1554-1568.

FLOSS. "Free/Libre and Open Source Software: Survey and Study," International Institute of Infonomics, University of Maastricht, The Netherlands, 2002 (available online at **http://www.infonomics.nl/FLOSS/report**; accessed February 15, 2004).

Fornell, C., and Larcker, D. "Evaluating Structural Equation Models with Unobserved Variables and Measurement Error," *Journal of Marketing Research* (18), 1981, pp. 39-50.

Gemser, G., and Wijnberg, N. "Effects of Reputational Sanctions on the Competitive Imitation of Design Innovations," *Organization Studies* (22:4), 2001, pp. 563-591.

Ghoshal, S., and Moran, P. "Bad for Practice: A Critique of the Transaction Cost Theory," *Academy of Management Review* (21:1), 1996, pp. 13-47.

Granovetter, M. S. "The Strength of Weak Ties," *American Journal of Sociology* (91), 1973, pp. 481-510.

Hartwick, J., and Barki, H. "Explaining the Role of User Participation in Information System Use," *Management Science* (40:4), 1994, pp. 440-465.

Hayes, M., and Greenemeier, L. "Oracle's Infrastructure Now Fully Linux-ized," *Information Week*, August 7, 2003 (available online at **http://www.informationweek.com/story/showArticle.jhtml?articleID=13000344**; accessed May 4, 2004).

Ives, B., Olson, M. H., and Baroudi, J. J. "The Measurement of User Information Satisfaction," *Communications of the ACM* (26:10), 1983, pp. 785-793.

Janz, B. D., Wetherbe, J. C., Davis, G. B., and Noe, R. A. "Reengineering the Systems Development Process: The Link between Autonomous Teams and Business Process Outcomes," *Journal of Management Information Systems* (14:1), Summer 1997, pp. 41-68.

Jones, C., Hesterly, W. S., and Borgatti, S. P. "A General Theory of Network Governance: Exchange Conditions and Social Mechanisms," *Academy of Management. The Academy of Management Review* (22:4), October 1997, pp. 911-945.

Kanhey, L. "Open Source Gurus Trade Jabs," *Wired*, April 10, 1999 (available online at **http://www.wired.com/news/technology/0,1282,19049,00.html**; accessed March 25, 2004).

Kraut, R. E., and Streeter, L. A. "Coordination in Software Development," *Communications of the ACM* (38:5), 1995, pp. 69-81.

Lakhani, K. R., Wolf, B., Bates, J., and DiBona, C. "The Boston Consulting Group/OSDN Hacker Survey," The Boston Consulting Group, January 2002 (available online at **http://www.bcg.com/opensource/BCGHACKERSURVEY.pdf**; accessed December 4, 2003).

Markus, M. L., Manville, B., and Agres, C. E. "What Makes a Virtual Organization Work?," *Sloan Management Review* (42:1), Fall 2000, pp. 13-26.

Mims, B. "Novell, IBM and HP Unite Efforts to Put Linux on Top," *Salt Lake Tribune*, Salt Lake City, March 25, 2004, Business Section, p. E1.

Mockus, A., Fielding, R. T., and Herbsleb, J. D. "Two Case Studies of Open Source Software Development: Apache and Mozilla," *ACM Transactons on Software Engineering Methodologies* (11:3), 2002, pp. 309-346.

Moore, G. C., and Benbasat, I. "Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation," *Information Systems Research* (2), 1991, pp. 192-222.

Netcraft.net. "Netcraft Web Server Survey," 2004 (available online at **http://news.netcraft.com/archives/2004/05/03/may_2004_web_server_survey_finds_50_million_sites.html**; accessed Mary 2, 2004).

Olson, M. *The Logic of Collective Action*, Harvard University Press, Cambridge, MA, 1965.

Pinto, M. B., Pinto, J. K., and Prescott, J. E. "Antecedents and Consequences of Project Team Cross-Functional Cooperation," *Management Science* (39:10), October 1993, pp. 1281-1297.

Raymond, E. S. "The Cathedral and the Bazaar," 2000 (available online at **http://www.catb.org/~esr/writings/cathedral-bazaar/**; accessed September 20, 2003).

Sawyer, S. "A Social Analysis of Software Development Teams: Three models and Their Differences," in *Proceedings of the Sixth Americas Conference on Information Systems*, M. Chung (Ed.), Long Beach, CA, 2000, pp. 1645-1649.

Sawyer, S., and Guinan, P. J. "Software Development: Processes and Performance," *IBM Systems Journal* (37:4), 1998, pp. 552-569.

SourceForge.net. "SourceForge.net: Top Forum Post Counts," 2004a (available online at **http://sourceforge.net/top/toplist.php?type=forumposts_week**; accessed January 1, 2004).

SourceForge.net. "SourceForge.net: Welcome," 2004b (available online at **http://sourceforge.net/**; accessed January 1, 2004).

Taylor, W. C. "Inspired by Work," *Fast Company*, November 1999, p. 200 (available online at **http://www.fastcompany.com/magazine/29/inspired.html**; accessed April 27, 2004).

Werts, C. E., Linn, R. L., and Joreskog, K. G. "Intra-class Reliability Estimates: Testing Structural Assumptions," *Educational and Psychological Measurement* (34), 1973, pp. 25-33.

Williamson, O. E. "Opportunism and its Critics," *Managerial and Decision Economics* (14:2), Special Issue: Transactions Costs Economics, March-April 1993, pp. 97-107.

Williamson, O. E. (Ed.). *Transaction Cost Economics and Organization Theory*, Princeton University Press, Princeton, NJ, 1994.

Williamson, O. E., and Winter, S. G. *The Nature of the Firm* (2nd ed.), Oxford University Press, Oxford, 1993.

Wold, H. "Systems Under Indirect Observation Using PLS," in *A Second Generation of Multivariate Analysis*, C. Fornell (Ed.), Praeger, New York, 1982, pp. 325-347.