Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2008 Proceedings

Americas Conference on Information Systems (AMCIS)

2008

# Impacts of Social Network Structure on Knowledge Sharing in Open Source Software Development Teams

Yuan Long
*Colorado State University Pueblo*, yoanna.long@colostate-pueblo.edu

Keng Siau
*Univesity of Nebraska Lincoln*, siauk@mst.edu

# Impacts of Social Network Structure on Knowledge Sharing in Open Source Software Development Teams

**Yuan Long**
Colorado State University Pueblo
Yoanna.long@colostate-pueblo.edu

**Keng Siau**
University of Nebraska Lincoln
ksiau@unl.edu

## ABSTRACT

The study examines the relationship between social network structure and knowledge sharing in Open Source Software (OSS) development teams. One hundred and fifty projects were selected from SourceForge.net using stratified sampling. Social network structure was measured by two indices: degree of centralization and core/periphery fitness. Knowledge sharing was measured from two aspects: the quality of knowledge sharing that is indicated by the helpfulness of messages and the quantity of knowledge sharing that is indicated by the number of messages. The results show that social network structure significantly affects the quantity of knowledge sharing. However, social network structure does not influence the quality of knowledge sharing. In addition to the contribution to OSS literature, the results of this study also inform OSS practice.

### Keywords

Open Source Software (OSS) development, social network, network structure, knowledge sharing, content analysis

## INTRODUCTION

Open Source Software (OSS) development has received increasing attention from both researchers and practitioners in the past few years. OSS refers to the computer software that opens its source code to public (OpenSourceInitiative, 2006). Anyone who is interested in the software is able to access, modify, customize, and redistribute the software under an open source license (OpenSourceInitiative, 2006).

Most of the participants of OSS projects are volunteers, distributed all over the world. They communicate and collaborate with each other through the Internet using tools such as discussion forum and mailing list. They form a virtual community, in which individual personalities and organizational attributes become less significant while social relationships and network structures exhibit stronger influences on behavior. In such a community, "knowing who knows what" is as important as, if not more important than, "knowing what" and "knowing how" in (Borgatti, 2003). The phenomenon leads to increasing research interests regarding the influences of external social relationships on collaboration behaviors in virtual teams (Burt, 1992; Putnam, 1995).

In addition to social relationships, effective and efficient knowledge sharing is critical to OSS success. By using social network analysis and computer-aided content analysis, this paper examines the relationship between social network structure and knowledge sharing in OSS development teams. The paper is organized as follows: the next section presents a literature review on social network and knowledge sharing. The following section introduces the research methods including sample selection, data collection, and measurement. Research results are then provided, followed by discussions and future research directions.

## LITERATURE REVIEW

### Social network structure in virtual communities

Researches in information processing and organizational learning have consistently argued for the importance of social networks in group practices, especially knowledge processes (Cross, 2001). The focus of social network research can be either on the dyadic relationship between two individual units or on the structure of the entire network. Focusing on the group level, our study investigates the social structure of the entire network.

Social network structure relies on a set of properties (e.g., *density*, *centralization*, and *hierarchy*) that defines the structural pattern of the network as a whole. Different knowledge practices may require different social structures. For example, Borgatti (2005) argued that dense networks promote knowledge sharing but hinder most knowledge creation at the same time. Among all the other properties, centralization and core/periphery are the two important traits of social network structure.

#### *Degree of Centralization*

Centralization refers to the extent to which a group revolves around a single center. In other words, how tightly the graph is organized around its central point (Scott, 2000). The centralization is measured by the aggregate differences between the centrality score of the most central point and scores of the other points. Centralization is the ratio of the actual sum of differences to the maximum possible sum of differences (Scott, 2000).

The group centralization ranges from 0 to 1. 1 indicates a completely centralized network (such as a star) while 0 indicates a completely decentralized network.

#### *Core/periphery Fitness*

The core/periphery fitness measures the extent to which the network is close to a perfect core/periphery structure. In a typical core/periphery structure, a closure core exists with a loose periphery (Borgatti, 2002). The core encompasses key group members with strong ties between each other while the periphery contains members that are usually connected to the core through weak ties.

Core/periphery fitness is calculated by assessing the closeness between a data network and a perfect core/periphery network (Borgatti, 2002). The value of core/periphery varies from 0 to 1. The higher the index, the closer the network looks like a core/periphery structure.

### Knowledge sharing in virtual communities

Knowledge has been recognized as the primary resource of organizations (Alavi, 2001; Argote, 2003). The theoretical explanations of knowledge practices can be generally summarized into two broad categories: properties of units and properties of relationships between units (Argote, 2003).

Considerable research has been conducted to explore the influence of the properties of units on knowledge practices. However, research regarding the impact of the relationships between units on knowledge transfer is relatively new (Argote, 2003). Earlier literature suggested a relationship between underlying social mechanism and knowledge practices (Tsai, 2002; Wasko and Faraj, 2005; Borgatti and Cross, 2003; Thomas-Hunt, 2003; Kankanhalli, 2005). However, how certain social network factors affect knowledge process is not evident; more work needs to be done to provide insights in this area.

Group centralization can be viewed as a measure of inequity between individual actors (Wasserman, 1994). A centralized structure reveals an uneven distribution of knowledge where knowledge is concentrated in the center (Ahuja, 1999). Moreover, a centralized structure largely depends on a single leader who may not always be capable to provide sufficient support for other group members. Furthermore, exchanging knowledge and maintaining contact with many others by a single center would be time consuming; and the high cost may reduce the efficiency and effectiveness of knowledge communication (Burt, 1992). Therefore, the study hypothesizes that the degree of centralization is negatively related to knowledge sharing in OSS teams.

*H1a: The degree of centralization has a negative relationship with the quantity of knowledge sharing in OSS teams.*
*H1c: The degree of centralization has a negative relationship with the quality of knowledge sharing in OSS teams.*

In open source software development, a project is often initiated by several programmers who share common interests. Maintaining a group of key developers is critical to the success of the project. This group of developers makes the majority of the contributions and motivates others to engage in the development. They take lead on discussion, reply queries, maintain a regular interaction with testers and users, and keep the project active.

In addition, according to the literature (Krebs and Holley, 2004), most of the organizations start from a scattered structure to a core/periphery network in the end, which lead to efficient performance.

Therefore the study hypothesizes that greater core/periphery fitness will be positively related to group performance.

*H1b: The core/periphery fitness has a positive relationship with the quantity of knowledge sharing in OSS teams.*
*H1d: The core/periphery fitness has a positive relationship with the quality of knowledge sharing in OSS teams.*

## RESEARCH METHOD

### Sample Selection

Project samples were selected from SourceForge.net, which is the world's largest website to host OSS projects[1]. Stratified sampling was used in the study. The strata were determined by the software categories provided by SourceForge. In each software category, the size of the sample is taken in proportion of the stratum. Table 1 lists the descriptive information of the project samples.

| Software Category | Number of Projects Sampled | Number of Developers | | | Project Tenure[1] (Month) | | |
|---|---|---|---|---|---|---|---|
| | | Min | Max | Mean[2] | Min | Max | Mean[2] |
| Database | 8 | 5 | 91 | 7 | 20 | 69 | 55 |
| Games/Entertainment | 20 | 3 | 68 | 20 | 17 | 72 | 48 |
| Communication/Internet | 30 | 4 | 325 | 35 | 7 | 74 | 53 |
| Office business | 17 | 3 | 46 | 13 | 7 | 69 | 37 |
| Scientific/Engineering | 31 | 2 | 45 | 12 | 6 | 72 | 42 |
| Software development | 22 | 2 | 55 | 17 | 11 | 72 | 45 |
| System | 11 | 2 | 36 | 18 | 16 | 74 | 43 |
| Others | 11 | 3 | 67 | 21 | 26 | 74 | 57 |
| **Summary** | **150** | **2** | **325** | **21** | **6** | **74** | **47** |

[1] Counted from the project registered date to the date of the study
[2] Rounded to nearest number

**Table 1. Descriptive information for sample**

### Data Collection

The data were gathered from the bug tracking system of each selected project. The bug tracking system enables participants, including both developers and users, to report, discuss, and track bugs. In the bug tracking system, somebody reports a bug and others reply to the message with discussions and solutions.

We chose bug-tracking system as the primary data source for the following reasons. First, OSS development is characterized as peer review of open codes. Raymond (2000) proposed the "Linux' law" in his well known essay "Cathedral and the Bazaar", "Given enough eyeballs, all bugs are shallow." Therefore, the bug tracking system is a key feature of OSS development as well as an important venue for participants (both developers and users) to collaborate with each other. Second, compared to other development activities such as feature request, the bug-fixing process is the most active procedure that reveals close collaboration and rich interaction. Third, the majority of the posts on bug tracking systems center around bug reports and bug solving. The topics are much more focused than other forums such as mailing lists. In mailing lists, people can chat on whatever topics they desire, which may not be relevant to knowledge sharing in software development. Lastly, most of the OSS projects open their bug tracking systems to the public, which can be accessed by academic researchers. These bug tracking systems record the historical threads since the beginning of the projects. Some other communication tools such as mailing lists are not always open to the public but only to those registered participants. In addition, group meetings through MSN messenger[2] or Skype[3] are neither recorded nor open to the public.

In this study, the entire bug-report Web pages of each selected project were downloaded. A total of 76,992 bug reports were incorporated in the study, with an average of 513 bug reports per project. Each Webpage (a bug report) includes multiple messages, which were used as data source for the content analysis.

### Social Network Structure measurement

Social Network Analysis (SNA) was used to analyze the social structures of OSS teams. SNA is characterized as a distinctive methodology encompassing techniques for data collection, statistical analysis, and visual representation (Katz, 2004). To

[1] As reported on Feb. 29, 2008, SourceForge hosted over one hundred thousand projects and involved over one million registered users (170,945 projects and 1,801,545 registered users).
[2] MSN messenger is an Internet instant messenger service (Wikipedia)
[3] Skype is a peer-to-peer telephony network (Wikipedia)

generate the interactive data from the bug tracking system in the study, the study followed a three-step procedure as shown in Table 2.

| Steps | Summary | Software | Input | Output |
|---|---|---|---|---|
| Step 1. Web page download | To download all the bug report pages from each selected project | Web spider program[1] | Bug tracking system on SourceForge | Bug report web pages |
| Step 2. Web page analysis | To generate a matrix revealing the interaction among users for each project | Web parsing program[1] | Bug report web pages | A matrix for each project |
| Step 3. Statistical analysis | To calculate the indices of network structure | Ucinet[2] | Matrix | Indices such as centrality and core/periphery fitness |

[1] Programs were developed based on the work of Crowston and Howison ( 2005) with necessary revision
[2] Software was developed by Borgatti, Everett, and Freeman (2002)

**Table 2. Social Network Analysis Process**

**Step 1 Web page download**

The bug report pages from each project were downloaded using a web spider program. The bug report pages served as the data source for the following analyses.

**Step 2 Web page analyses**

After downloading the bug report pages, a social matrix was generated for each project to reveal the interaction among users. A Web parsing program was used to produce this matrix (an example is shown in figure 1). The rows and the columns represent actors in the social network, which were determined by the unique IDs of each OSS project. The cells represent the interactions among actors, which were determined by the number of replies between a pair of actors.

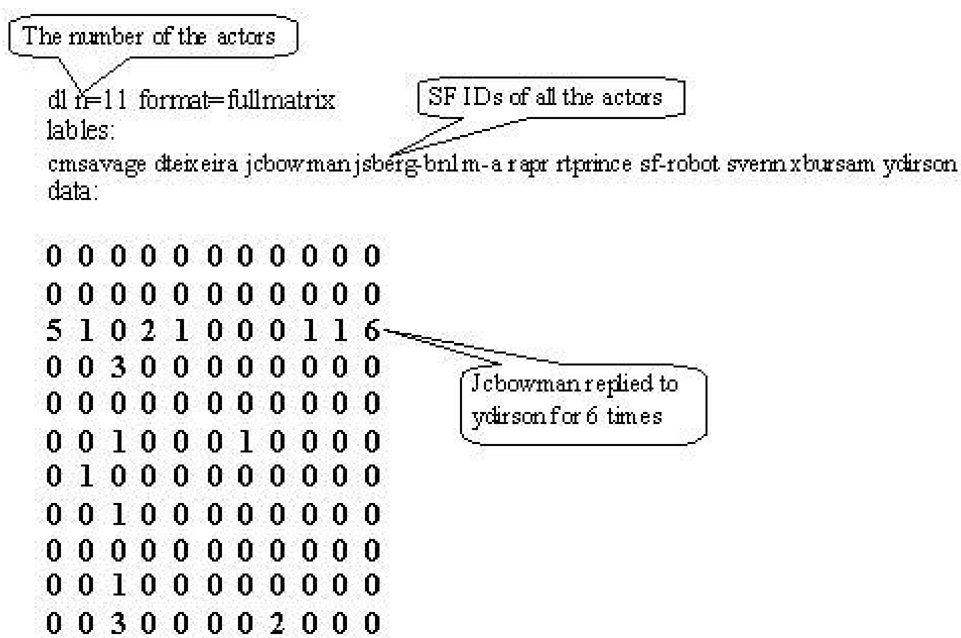


**Figure 1. an Example of the Generated Matrix**

**Step3. Statistical Analysis**

The matrices generated from step 2 served as input for the SNA software program. In the study, UCINET (Borgatti, 2002) was used to calculate the indices of network structure. As mentioned earlier, the study focuses on two important indices: the degree of centralization and core/periphery fitness.

### Knowledge sharing measurement

*Computer aided content analysis*

The extent of knowledge sharing of each message was coded using computer-aided content analysis software. Content analysis is a research method in social science to analyze communication content (Babbie, 2006; Holsti, 1969; Weber, 1990). It has been actively used in different fields for more than 50 years to determine the presence of certain words or concepts in texts (Wagner et al., 2003; Busch et al., 2005).

*Measurement*

The extent of knowledge sharing was measured from two aspects: the quality and quantity of knowledge sharing.

Quality of knowledge sharing

The quality of messages is indicated by the degree of helpfulness of messages. Content analysis was employed to determine the helpfulness of messages in terms of three levels: very helpful, somewhat helpful, and not at all helpful. The coding policy is based on the work of Wasko and Faraj (2005). These three levels of helpfulness are defined as below.

- **Very helpful** (coded as 2). The response does not only answer the question directly, but also provides explanations or references to relevant knowledge sources.
- **Somewhat helpful** (coded as 1). The response answers the question but provides little explanations. Another possibility is that the response does not answer the question directly but provides information relevant to the problem.
- **Not helpful** (coded as 0). The response is not related to the query. It can be a question itself, a social greeting (such as *Thank you* and *Hello*), or an announcement. In any of these cases, the response is not helpful in sharing knowledge.

The coding process is shown in Figure 2.

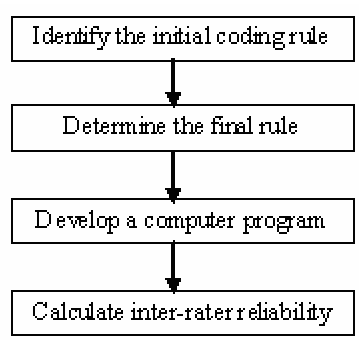


**Figure 2. Coding Process**

**<u>Step 1 Identifying the Initial Coding Rule</u>**

The first step is to identify the coding rule, which is guided by the above coding policy (e.g., three levels of helpfulness of messages). One hundred bug tracking pages were randomly selected (over 200 messages in total) from the sampling pool. Two human coders, who are experts in software development, coded each message independently based on the coding policy. In addition to rating, the two coders documented the reason for every decision. After carefully comparing the coding results and discussing the discrepancies, they agreed on an initial coding rule. The rule comprise of two parts: a set of key words and the length of the message.

There are two reasons for choosing both the keywords and the length of the messages to assess the helpfulness of each message. First, the discussion board (i.e., bug tracking systems) is a specific space for IT professionals to discuss bugs. The topics covered are much more focused than some other social forums. Therefore, it is feasible to identify a set of key words which appear frequently and are closely related to bug tracking such as add, create, and CVS.

However, the set of key words may not be sufficient to code all the messages that contribute to bug tracking. For example, a long paragraph of revised coding posted by a programmer may not include any keywords. Therefore, the length of a message is another component to assess the helpfulness of the messages.

**Step 2 Determining the Final Rule**

In order to ensure the accuracy of the coding rule, the two coders applied the initial coding rule to rate another one hundred bug tracking pages which were randomly selected from the sampling pool. They continuously revised the initial rule until they reached agreement.

An inter-rater reliability between the two human coders was calculated after the coding. The reliability was accessed at 0.87 using Krippendorff's alpha statistic (Krippendorff, 1980) and at 0.75 using Kappa's alpha statistic, which indicate a relatively high reliability of the coding. The final coding rule is based on the following matrix shown in Table 3. For example, if a message includes no key words and the length of the message is less than 20 words, it is coded as 0.

| K.W.[2] \ L.M.[1] | 0-20 | 20-50 | >50 |
|---|---|---|---|
| 0 | 0[3] | 0 | 1 |
| 1 | 1 | 1 | 2 |
| >=2 | 1 | 2 | 2 |

[1] L.M.: Length of the Message
[2] K.W.: Number of keywords in the message
[3] Coding. For example, if the length of a message is less than 20 words and contains no keywords, it will be coded as 0.

**Table 3. Coding Rule**

**Step 3 Developing a Content Analysis Program**

After determining the coding rules, the next step is to integrate the rule into a computer program. A content analysis program was developed by one author using Perl (Practical Extraction and Report Language). Perl is free software and has been widely used in text related tasks.

**Step 4 Calculating the Inter-rater Reliability**

To ensure the validity and reliability of the content analysis program, the two coders and the computer program coded 100 bug tracking web pages that were randomly selected from the sampling pool. Minor revisions were made to solve the discrepancy between the two coders as well as between the coders and the computer program.

Table 4 shows the inter-rater reliability between the two coders as well as between the coders and the computer program. The results indicate a relatively high reliability of the coding process.

| | Coder 1 | Coder 2 | Computer program |
|---|---|---|---|
| Coder 1 | | | |
| Coder 2 | 0.87/0.72[1] | | |
| Computer program | 0.72/0.68 | 0.71/0.68 | |

[1] Krippendorff's alpha / Kappa's alpha

**Table 4. Inter-rater Reliability**

### Quantity of knowledge sharing

The second measure of knowledge sharing assesses the volume of messages, which indicates the quantity of knowledge sharing. The value was calculated using the mean number of messages (the number of messages posted minus the number of messages coded as 0) posted by the whole group for each month.

Table 5 summarizes the measured variables of network structure, knowledge sharing, and control variables.

| | | Brief definition | Measurement |
|---|---|---|---|
| Network Structure | Centralization | The extent to which a network revolves around a single node | Social network analysis |
| | Core/periphery fitness | The extent to which a dense core exists together with a sparse periphery in a network | |
| Knowledge Sharing | Quantity | The number of messages | Content analysis |
| | Quality | The degree of helpfulness of the messages in terms of knowledge sharing | |
| Control Variables | Group size | The number of developers | Archival analysis |
| | Project type | Software categories in terms of the major function provided by that software | |
| | Project tenure | The number of months from the start of the project until the date of the analysis | |
| | Project size | The Source Lines of Code (SLOC) of OSS package for each project | |

**Table 5. Measurement of the Variables**

## RESEARCH RESULTS

Table 6 shows the descriptive statistics for the variables in this study. Table 7 shows the statistic results, in which knowledge sharing is the dependent variable.

To test the effect of network structure on the quantity of knowledge sharing, the first step is to include only the control variables (model 1). The two network structure variables (model 1') are then added. The statistical results show that the R-square change is significant at 0.01 level, which means that network structure significantly predicts the quantity of knowledge sharing. In addition, the coefficient of the degree of centralization is negative and significant, which supports the hypothesis that the higher the degree of centralization, the lower the volume of knowledge sharing. The coefficient of the core/periphery fitness is positive and significant, which supports the hypothesis that the higher the core/periphery fitness, the higher the volume of knowledge sharing. Therefore, both hypotheses 1a and 1b are supported.

As for the quality of knowledge sharing, neither of the two models (model 2 and model 2') is significant. This means the network structure variables together with the control variables do not affect the quality of knowledge sharing. In addition, the coefficients of both centralization and core/periphery fitness are not significant. Therefore, hypotheses 1c and 1d are not supported.

| | Variables | | Mean | S.D.[4] | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ | $X_7$ | $X_8$ | $X_9$ | $X_{10}$ | $X_{11}$ | $X_{12}$ | $X_{13}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C.V.[1] | $X_1$ | Project tenure | 46.6 | 18.5 | | | | | | | | | | | | | |
| | $X_2$ | Group size | 21 | 31 | 0.09 | | | | | | | | | | | | |
| | $X_3$ | Project size | 4.97 | 0.59 | 0.09 | 0.34** | | | | | | | | | | | |
| | $X_4$ | Database | 0.05 | 0.22 | 0.14 | -.08 | 0.12 | | | | | | | | | | |
| | $X_5$ | Game/ Entertainment | 0.14 | 0.35 | 0.02 | 0.01 | 0.05 | -.09 | | | | | | | | | |
| | $X_6$ | Internet/ Comm. | 0.20 | 0.40 | 0.14 | 0.22** | -.08 | -.12 | -.20* | | | | | | | | |
| | $X_7$ | Office | 0.12 | 0.32 | -.19*[5] | -.09 | -.12 | -.09 | -.15 | -.18* | | | | | | | |
| | $X_8$ | Scientific/ engineering | 0.21 | 0.41 | -.11 | -.13 | 0.04 | -.12 | -.21* | -.26** | -.19* | | | | | | |
| | $X_9$ | Software/ devp. | 0.14 | 0.35 | -.04 | -.05 | 0.01 | -.10 | -.16* | -.21* | -.15 | -.21** | | | | | |
| | $X_{10}$ | Systems/ security | 0.07 | 0.26 | -.05 | -.02 | -.06 | -.07 | -.11 | -.14 | -.10 | -.14 | -.12 | | | | |
| N.S.[2] | $X_{11}$ | Centralization | 0.13 | 0.11 | -.35** | -.30** | -.10 | -.15 | -.14 | -.24** | 0.02 | 0.51** | 0.01 | -.06 | | | |
| | $X_{12}$ | Core/periphery | 0.69 | 0.13 | 0.04 | -.08 | 0.14 | -.15 | -.07 | -.15 | -.02 | 0.25** | 0.08 | -.08 | 0.12 | | |
| K.S.[3] | $X_{13}$ | Quality | 1.2 | 0.16 | 0.21* | 0.05 | 0.02 | 0.13 | -.08 | 0.02 | 0.08 | -.01 | 0.05 | -.05 | -.12 | .05 | |
| | $X_{14}$ | Quantity | 57.4 | 34.1 | 0.02 | 0.44** | 0.24** | 0.03 | 0.14 | 0.08 | -.14 | -.23** | 0.06 | -.03 | -.42** | .09 | -.04 |

[1]C.V. Control variable; [2] N.S. Network Structure; [3]K.S. Knowledge Sharing; [4] S.D. Standard deviation; [5]**P<0.01,*P<0.05

**Table 6. Descriptive statistics of the variables**

| Predictors | D.V. Quantity of knowledge sharing | | D.V. Quality of knowledge sharing | |
|---|---|---|---|---|
| | Control variables (Model 1) | Network structure (Model 1') | Control variables (Model 2) | Network Structure (Model 2') |
| Project tenure | -.128 | -.239** | .245** | .217* |
| Group size | .392** | .332** | .043 | .031 |
| Project size | .090 | .053 | -.004 | -.015 |
| Database | -.116 | -.106 | .212* | .217* |
| Game | -.047 | -.056 | .112 | .111 |
| Internet | -.192 | -.195 | .202 | .203 |
| Office | -.271* | -.268* | .289* | .290* |
| Scientific | -.365** | -.235 | .244 | .272 |
| Software Devp. | -.118 | -.105 | .256 | .258 |
| Systems | -.158 | -.158 | .105 | .106 |
| Centralization | | **-.373**** | | **-.087** |
| Core/periphery | | **.165*** | | **.054** |
| $R^2$ | 0.289 | 0.398 | 0.102 | 0.110 |
| $R^2$ change | 0.289** | 0.109** | 0.102 | 0.007 |
| F | 5.663** | 7.564** | 1.583 | 1.405 |

Note: *<0.05 (2-tailed), **<0.01 (2-tailed)

**Table 7. Statistical Results of Regression Analysis (knowledge sharing as the D.V.)**

## DISCUSSIONS

The research results indicate that both the attributes of social network structure (i.e., the degree of centralization and core/periphery fitness) significantly affect the quantity of knowledge sharing in OSS teams. The higher the degree a group revolves around a single center, the lower the number of the messages shared among group members. High centralization indicates formal hierarchical structure, which may not be an efficient structure in OSS teams. The one-single-center structure is efficient when the initiator (the leader) starts the project. However, as more and more people join the project, the cost of maintaining the original one-single-center structure may increase.

The research results show that the greater the core/periphery structure exhibited by an OSS team, the higher the number of messages shared among group members. The core/periphery structure involves a group of core members, who maintain the stability of the teams and play key roles in the project development. The peripheries mainly include users and testers, who communicate and collaborate with the core members on a regular basis. Users and testers also contribute to the software development significantly. Users are the "eyeballs" to make the bugs "shallow" (Raymond, 2000) and can be treated as "co-developers for rapid code improvement and effective debugging" (Raymond, 2000). Core/periphery would be an effective way to support collaboration in OSS teams.

Although the results show a significant relationship between social network structure and quantity of knowledge sharing, they do not support similar relationship between social network structure and quality of knowledge sharing. One explanation would be that the quality of knowledge sharing depends more on the individuals' knowledge background rather than the social structure of OSS teams. For example, suppose two groups both have similar core/periphery structure. The members in the first group are mostly experts while the members in the second group are mostly novices. It is likely that the first group would share higher quality knowledge than the second one. Therefore, the network structure affects the amount of information people shared but may not influence the quality of the shared information.

## CONTRIBUTIONS AND FUTURE RESEARCH DIRECTIONS

The research examines the relationship between social network structure and knowledge sharing in OSS teams. Social network structure was measured by two structural attributes: the degree of centralization and core/periphery fitness. The knowledge sharing was measured from two aspects: the quality of knowledge sharing that is indicated by the helpfulness of messages, and the quantity of knowledge sharing that is indicated by the volume of messages. A computer program was developed to automatically code the helpfulness of messages based on a set of keywords and the length of the messages. Inter-rater reliability was calculated to ensure the reliability of the coding method.

The research has significant implications to the IS literature on social network, knowledge management, and virtual groups.

First, the study highlights the importance of social networks in OSS success. Software development is an intelligence-intensive process. The success of OSS projects usually does not depend on a single genius developer, but on the collaboration among developers with different specialties as well as between developers and users who continuously test the coding. Comparing to previous studies that focus on the internal attributes of the individuals, this research emphasizes the importance of social interactions in determining the group outcomes. The empirical evidence shows that the social network structure significantly affects the quantity of knowledge sharing.

Second, examining knowledge sharing among participants of OSS projects has important implications for research in knowledge management. Although considerable studies have research on knowledge practice in group settings, few of them systematically investigate the underlying social process of knowledge diffusion from the perspective of social network. The results of this research advance the understanding of the underlying social mechanism of knowledge sharing. Moreover, it provides evidence showing that knowledge sharing is a social process requiring social interactions among participants of OSS teams.

In addition to IS literature, the research has significant implications to IS practice. The results suggest guidelines and methods to construct an efficient social network to promote OSS performances. First, the research results emphasize the importance of maintaining a group of core members in OSS development. Second, the research suggests keeping regular relationships between key developers and those in the periphery. Third, the results caution the development of a highly centralized structure. A centralized structure depends highly on a single center. Without distributing tasks and sharing responsibilities, an OSS project is hard to go forward.

Future research should focus on two directions. One is to further improve on the measurement of knowledge sharing in OSS teams. There are some other communication tools in OSS teams, such as mailing lists and discussion forums. Future research would analyze the data from those forums and compare different communication patterns and knowledge sharing processes. In addition, other measurement techniques such as survey may be employed to complement the direct coding of messages. Second, future research should study the consequences of knowledge sharing in OSS development. It is important to understand the impact of knowledge sharing and social network structure on group performance.

## REFERENCES


1. Ahuja, M. K., Carley, K. M., "Network Structure in Virtual Organizations." Organization Science **10**(6), 1999, pp. 741-757.
2. Alavi, M. Leidner, D. E. "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," *MIS Quarterly* (25:1), 2001, pp. 107-136.
3. Argote, L., McEvily, B. and Reagans, R. "Managing Knowledge in Organizations: An Integrative Framework and Review of Emerging Themes," *Management Science* (49:4), 2003, pp. 572-582.
4. Babbie, E. *The Practice of Social Research, 10th edition*, Wadsworth: Thomson Learning Inc, 2006.
5. Borgatti, S. P. "Creating Knowledge: Network Structure and Innovation." *http://www.socialnetworkanalysis.com/knowledge_creation.htm*. August 2005.
6. Borgatti, S. P., Everett, M.G. and Freeman, L.C. *Ucinet for Windows: Software for Social Network Analysis*. Harvard, MA, Analytic Technologies, 2002.
7. Borgatti, S. P., and Cross, R. "A Relational View of Information Seeking and Learning in Social Networks," *Management Science* (49:4), 2003, pp. 432-445.
8. Burt, R. S., *Structural Holes: the Social Structure of Competition*. Cambridge MA, Harvard University Press, 1992.
9. Busch, C., De Maret, P. S., Flynn, T., Kellum, R., Le, S., Meyers, B., Palmquist, P., Saunders, M., and White, R. "Content Analysis". *http://writing.colostate.edu/guides/research/content/*, 2005.
10. Cross, R., Ronald E. Rice, and Andrew Parker. "Information Seeking in Social Context: Structural Influences and Receipt of Information Benefits." *IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews,* 2001, 31(438-448).
11. Crowston, K., and Howison, J. "The social structure of Free and Open Source software development," *First Monday* (10:2), 2005.
12. Holsti, O. R. "Content Analysis for the Social Sciences and Humanities," *Reading*, Mass, 1969.
13. Katz, N., Lazer, D., Arrow, H. and Contractor, N. "Network Theory and Small Groups." *Small Group Research*, 2004, 35(3): 307-332.
14. Krebs, V. & Holley, J., "Building Sustainable Communities through Network Building." http://www.orgnet.com/BuildingNetworks.pdf. 2002, accessed in 2005.
15. Krippendorff, E. *Content Analysis: An Introduction to its Methodology*, Beverly Hills, CA: SAGE, 1980.
16. Long, Y. "Social Structure for Open Source Software Projects," *AMCIS 2005*, Omaha, NE, 2005.
17. Long, Y. and Siau, K. "Social Network Structures in Open Source Software Development Teams," *Journal of Database Management* (18:2), 2007, pp. 25-40.
18. OpenSourceInitiative, "The Open Source Definition (V.1.9)," *http://www.opensource.org/docs/definition.php*. July, 2006.
19. Putnam, R. D. "Bowing along: America's Declining Social Capital." *Journal of Democracy*, 1995, 6: 65-78.
20. Raymond, E. S. "The Cathedral and the Bazaar", *http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/*, 2000.
21. Scott, J. P. *Social Network Analysis: A Handbook*. London, SAGE, 2000.
22. Tsai, W. "Social structure of "coopetition" within a multiunit organization: coordination, competition, and intraorganizational knowledge sharing," *Organization science* (13:2), 2002, pp.179-190.
23. Wagner W. P., Chung, Q. B., and Najdawi, M. K. "The Impact of Problem Domains and Knowledge Acquisition Techniques: A Content Analysis of P/OM Expert System Case Studies," *Expert Systems with Applications* (24:1), 2003, pp. 79-86.
24. Wasko, M. M., and Faraj, S. "Why Should I Share? Examining Social Capital and Knowledge Contribution in Electronic Networks of Practice," *MIS Quarterly* (29:1), 2005, pp. 35-57.
25. Wasserman, S. Faust, K., Social Network Analysis. Cambridge, Cambridge University Press, 1994.
26. Weber, R. P., *Basic Content Analysis, 2nd ed.*, Newbury Park, CA: Sage, 1990.