KNOWLEDGE COORDINATION IN OPEN SOURCE SOFTWARE PROJECT TEAMS:

A TRANSACTIVE MEMORY SYSTEM PERSPECTIVE

APPROVED BY SUPERVISING COMMITTEE:

____
Glenn B. Dietrich, Ph.D., Chair

____
Yoris A. Au, Ph.D.

____
Jan G. Clark, Ph.D.

____
Don Lien, Ph.D.

Accepted: ____________________________________
Dean, Graduate School

DEDICATION

This dissertation is dedicated to my parents and sister. Without your unconditional support, love, and constant inspiration, I would not have been able to go through the most difficult time of my doctoral study.

KNOWLEDGE COORDINATION IN OPEN SOURCE SOFTWARE PROJECT TEAMS:

A TRANSACTIVE MEMORY SYSTEM PERSPECTIVE

by

XIAOGANG CHEN, M.S.

DISSERTATION
Presented to the Graduate Faculty of
The University of Texas at San Antonio
In partial Fulfillment
Of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY IN BUSINESS ADMINISTRATION

THE UNIVERSITY OF TEXAS AT SAN ANTONIO
College of Business
Department of Information Systems and Technology Management
May 2009

ACKNOWLEDGEMENTS

Pursuing a Ph.D. degree is a challenging yet valuable journey in my life. In this journey, I am so fortunate to have supports and encouragements from many wonderful, generous, and resourceful people to whom I'm indebted so much.

First and foremost, I will forever be indebted to my dissertation chair, Dr. Glenn B. Dietrich, who always had his office door open for me and encouraged me during my difficult times, and Dr. Jan G. Clark, who devoted countless hours to reading my dissertation and helping me improve it. Dr. Yoris A. Au and my outside committee member Dr. Don Lien provided support and contributed to the development of this dissertation by willingly sharing their insight, knowledge, and expertise.

Many other faculty members assisted me and deserve mention including Dr. Diane Walz who led me to the gate of the IS research field and Dr. Chino Rao who exposed me to a variety of IS research topics early in my Ph.D. education.

To all of my colleagues, who helped by contributing suggestions to the development of my dissertation, Darrell Carpenter, Nicole Beebe, Alex McLeod, Carlos Dorantes, Sriraman Ramachandran, Ruben Mancha, Humayun Zafar, Kenneth Johns, and Terri Davis. Your friendship and colleagueship are greatly appreciated.

May 2009

KNOWLEDGE COORDINATION IN OPEN SOURCE SOFTWARE PROJECT TEAMS: A TRANSACTIVE MEMORY PERSPECTIVE

Xiaogang Chen, Ph.D. in Business Administration
The University of Texas at San Antonio, 2009

Supervising Professor: Glenn B. Dietrich, Ph.D.

Although a large number of high-quality open source software (OSS) has been successfully produced, little is known about knowledge coordination in the OSS setting. Therefore, this dissertation investigates how the members of an OSS project team coordinate their knowledge of different domains to bear on software development tasks.

From the transactive memory system (TMS) perspective, this dissertation particularly examines antecedents of TMS and the relations among TMS, knowledge coordination behaviors of OSS developers, and their communication quality; furthermore, the study looks into the effects of knowledge coordination and communication quality on team performance.

By surveying 97 OSS project teams from Sourceforge.net, one of the largest OSS project hosting sites, the results of this dissertation demonstrate the importance of TMS for knowledge coordination behaviors and communication quality of the OSS developers. Moreover, communication quality shows the positive influence on team performance. These results contribute to the current literature as well as management practice.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE: INTRODUCTION

Problem Statement

The Open Source Software (OSS) phenomenon has generated much excitement in the software market in recent years. More and more companies are beginning to consider OSS as a viable and economic substitution for proprietary software to support their business processes. For example, the New York Stock Exchange (NYSE) recently adopted Linux to support its electronic trading platform due to its low cost, flexibility, and high level of security (Asay, 2008). In January 2008, Netcraft, an Internet services company based in England, surveyed 155,583,825 web sites worldwide. The results show that the Apache web server currently occupies 50.61 % of the web server market share (January 2008, www.netcraft.com). Quite a few software producing firms, such as Red Hat, VA Software, and Mozzilla have built their business models entirely on the OSS paradigm. Even Microsoft, the leading ideological opponent of the OSS community, launched its own OSS hosting site, CodePlex, in July 2006 (Voth, 2006).

Most OSS is developed and maintained by teams of voluntary developers scattered around the world (Crowston, Annabi, Howison, & Masango, 2004). These developers rarely, if ever, meet face-to-face and interact with each other almost exclusively through lean media (e.g., mailing-lists). Furthermore, most OSS project teams do not employ any "…traditional project coordination mechanisms such as formal planning, system-level design, schedules, and defined development processes" (Crowston et al., 2004, p. 18). Nonetheless, a number of high-quality, large-scale, complex software systems, such as Linux, Apache, and Perl, have been successfully produced through the OSS paradigm. This fact suggests that effective coordination must exist within some OSS project teams and some mechanisms must be employed to help coordinate behaviors of OSS developers in these teams. However, it remains largely unknown about what these coordination mechanisms are (Scacchi, 2001). Only a few researchers have looked into the issue. For example, Krogh, Spaeth, and Lakhani (2003) conducted a case study and found that developers in Freenet (i.e., an open source peer-to-peer file sharing software) coordinate with each other through participating mailing-list discussion and specializing in different modulations. Gutwin, Penner, and Schnerder (2004) interviewed fourteen developers of three highly successful OSS projects (i.e., NetBSD, Apache http, and Subversion) and concluded that developers of an OSS project team are able to form and maintain a broad awareness about who are working on what in the team. This broad awareness helps coordinate behaviors between developers.

Literature discussed above is focused on a few specific OSS cases, even though hundreds of others have demonstrated good quality (Stamelos, Angelis, Oikonomou, & Bleris, 2002). It is questionable whether the results from the case studies are generalizable to the large population of OSS projects. More importantly, the literature above has ignored an essential aspect of coordination in the software development context, that is, knowledge coordination. Knowledge is the most important resource for software development (Robillard, 1999), which typically involves a wide range of specialized knowledge, such as system design, programming, and business rules (Rus & Lindvall, 2002). Software development essentially is a knowledge coordination process in which developers purposely share and integrate their individually-possessed specialized knowledge to design a software solution for a business problem (Tiwana, 2004). By surveying 69 software development teams, Faraj and Sproull (2000) found that mere presence of specialized knowledge has no effects on team effectiveness (i.e., how well a team meets its project goals such as work quality, team operations, and design objectives) and efficiency (i.e., the team's adherence to its schedule and budget). Rather, the knowledge must be coordinated in order to produce positive impacts on team effectiveness and efficiency. Tiwana (2004) has similar findings. He theorized that software development requires two general types of knowledge: technical and application domain knowledge, and concluded that coordinating the two types of knowledge improves customer satisfaction and budget utilization, and decreases the number of bugs in software developed. Therefore, knowledge coordination is a key determinant of software development team performance.

However, little is known about knowledge coordination in the OSS setting (Gutwin et al., 2004; Mockus, Fielding, & Herbsleb, 2002) despite the fact that a large number of high-quality OSS has been successfully produced. No study has explicitly examined how OSS developers accomplish knowledge coordination although there are daunting barriers (e.g., no monetary incentive and geographic dispersion). Therefore, this dissertation intends to fill the gap in the literature by asking:

How do the members of an OSS project team coordinate their knowledge of different domains to bear on software development tasks?

The question above is of importance to study for several reasons: First, as OSS has increasingly become the integral component of software engineering, software engineers as well as IT managers want to learn from work practices of OSS project teams to "improve the effectiveness of software engineering as a human and team practice" (Crowston et al., 2004, p. 18). This research answers such a request by looking into the knowledge coordination practices of OSS project teams. Second, interest in the OSS phenomenon extends far beyond the software engineering field. Social scientists like IS researchers are deeply interested in coordination mechanisms of OSS project teams (Cusumano, 2005) and seek the possibility of applying the open source modes of coordination and organization to other areas (Ghosh, 2002).

Theoretical Perspective

As Crowston et al. (2004) points out, the current OSS research generally lacks a clear theoretical framework. Therefore, this research adopts theory of transactive memory system (TMS) as its theoretical lens to examine knowledge coordination behaviors of OSS developers. Wegner (1987) first conceived the concept of TMS to describe the cognitive interdependence in a group of people having close relationships (e.g., dating couples). In such relationships, the group members often rely upon each other as sources of "external memory storage" (Wegner, 1987, p. 187) to provide some group-relevant information. While the information itself is distributed among the different group members, the information about who knows what is commonly shared among team members. This interdependence results in a group-level "knowledge-holding system that is larger and more complex than" (Wegner, 1987, p. 189) any individual member's own memory system. Meanwhile, members can easily access the information stored in this system because the location of information is known. Wegner termed this knowledge-holding system the TMS and formally defined the TMS as a set of individual memory systems in combination with the shared awareness about information location among the group members.

Wegner reasoned that a TMS forms on the basis of knowledge responsibility. A group member can incur the responsibility for a certain knowledge domain if he or she is (1) perceived as the group's expert of the domain; (2) known to have access to knowledge of the domain; (3) or assigned by an authority to the domain (Wegner, 1987; Wegner, Erber, & Paula, 1991). Such a responsibility means that the group will channel to the member any new information related to the domain. The group will also consult the member when any questions related to the domain arise. As a result, this member becomes the source and repository of this knowledge domain for the group. Likewise, other group members might incur responsibilities of other knowledge domains, and hence, specialize in those domains. Eventually, a differentiated knowledge structure emerges within the group, where different experts in the group encode different domain knowledge.

Because each member holds differentiated knowledge, transactive integration is an essential process for a TMS to affect group performance (Wegner, 1987; Wegner, Giuliano, & Hertel, 1985). Transactive integration is an interactive cuing process, in which the knowledge provided by one member becomes the cue for other group members to retrieve relevant but different knowledge stored in their own memory systems. Integrating these knowledge pieces subsequently generates new knowledge that is qualitatively different from any single piece.

Wegner made an assumption that information flows freely within a group in his original theorization of TMS. This assumption is consistent with his early studies of dating couples but not necessarily true for other types of groups. However, this assumption should not be the concern for this research because an OSS project team typically consists of volunteers and they willingly contribute their time and expertise at no charge.

Extensive experimental and field studies have substantiated the importance of the TMS in a variety of group settings, such as dating couples, consulting teams, and new product development teams (Akgün, Byrne, Keskin, & Lynn, 2006; Austin, 2003; Hollingshead, 1998b; Lewis, 2004; Wegner et al., 1991). These studies found that the TMS enhances group performance through facilitating knowledge coordination within a group. Specifically, the TMS improves knowledge coordination in the following ways: 1) The TMS divides knowledge responsibilities among the members so that the group's cognitive resource is better utilized (Hollingshead, 1998a, 1998b; Lewis, 2004); 2) the TMS helps identify the location of knowledge within the group. Therefore, the members have "a quick and coordinated access to" (Lewis, 2004, p. 1519) needed task-relevant knowledge; 3) knowing each other's expertise enables the members to reasonably anticipate one another's knowledge needs and thus plan their own behaviors accordingly (Moreland, 1999; Moreland, Argote, & Krishnan, 1996).

As the above discussion shows, TMS has particular focus on explaining how knowledge is coordinated within a team. Therefore, this study adopts TMS as the theoretical perspective to research knowledge coordination behaviors in an OSS project team.

Scope of the Study

Two types of OSS project teams are prevalent in practice: "typical-OSS" and "semi-OSS." A "typical-OSS" project team has three defining characteristics: 1) The source code of the software that the team develops must be distributed under one of the copyleft licenses recognized by the OSS community (e.g., the GPL, the LGPL, and the MIT licenses); 2) the team's major workforce consists of voluntary developers who do not receive monetary incentives from corporate sponsors; and 3) team members coordinate their activities almost exclusively through the standard OSS collaboration tools (e.g., mailing-lists, online-discussion forums, and issue tracking systems). "Semi-OSS" projects, on the other hand, are often initiated by commercial software firms to pursue business opportunities. The exemplars of such OSS projects include Mozzilla, OpenOffice, and Red Hat Enterprise Linux. Besides voluntary code contributors, the "semi-OSS" projects have dedicated developers, often hired specifically, to work on the projects. Therefore, this type of OSS project teams has substantially contextual differences from the "typical-OSS" project teams. Such differences make the answer for the research question discussed above quite different between "typical-OSS" and "semi-OSS." For instance, "semi-OSS" projects have to consider what effect sponsorship has brought to knowledge coordination, whereas "typical-OSS" projects do not. This study confines itself to surveying and observing the "typical-OSS" project teams.

Organization of the Study

Chapter Two presents reviews of literature related to OSS, knowledge management, and TMS. Chapter Three discusses the research model and hypotheses developed under study. Chapter Four describes the research methodology. Chapter Five presents the results of the study. Chapter Six discusses the theoretical and pragmatic implications of the findings, the limitations of the research, and suggests future research directions.

CHAPTER TWO: LITERATURE REVIEW

The primary purpose of this study is to substantiate the importance of the TMS for knowledge coordination in the OSS settings. Thus, this chapter reviews prior research related to OSS, knowledge management, and TMS.

OSS Research

The OSS phenomenon has caught the eyes of many IS researchers in the past several years (Krogh & Hippel, 2006; Mockus et al., 2002). Consequently, a large number of OSS related studies have emerged, mainly focused on three strands – the motivations of OSS developers, the commercial implications of OSS, and the governance and organization of OSS development process (Crowston et al., 2004; Krogh & Hippel, 2006). This study is particularly related to the OSS development process. Therefore, the following review focuses on relevant literature in this area.

Organizational Structure of OSS Project Teams

It might appear at first sight that OSS teams are anarchic due to the volunteer nature of the workforce (Lerner & Tirole, 2002). However, centralization and hierarchy are prevalent in many OSS teams. For instance, in the Linux kernel project, Linus Torvalds and a group of delegates are responsible for reviewing the code other developers submit, and have the *de facto* authority to decide whether or not to include the code in the official version of Linux. Several OSS researchers, therefore, hypothesize that an ideal OSS project team should have a small number of core developers, a large number of co-developers, and an even larger number of beta-testers (e.g., Cox, 1998; Moon & Sproull, 2000; Raymond, 2000; Crowston & Howison 2006). Beta-testers use the software in their local computing environments and report any bugs encountered. Co-developers, based on bug reports, suggest possible code fixes. Then core developers make decisions about whether or not a certain fix is appropriate and should be included in the next release of the software.

Findings in recent empirical work support such a hypothesis. Ghosh and Prakash (2000) studied the pattern of code contributions in 3,149 OSS projects and found that the majority of the developers made only one contribution to their projects, and that a small portion of the developer population produced most of the coding. Mockus et al. (2002) have similar conclusions from their case study of Apache and Mozzilla development teams and discovered that each development team was actually comprised of two subgroups – the core group and an ancillary group. The core team was responsible for developing new functionalities, assigning work among developers, and inspecting the submitted bug fixes, whereas the ancillary group tested the released code in a variety of platforms and reported problems if any surfaced. Based on empirical data from both projects, Mockus et al. (2002) deduced that the core of developers should not be more than 10 to 15 developers; however, the ancillary group should have as many members as possible.

Communication and Coordination within an OSS Project Team

The OSS setting poses serious communication and coordination barriers for OSS developers. For example, developers working on the same OSS project may never meet each other face-to-face, may have different cultural backgrounds, and may live in different time zones. However, the high quality and wide user-acceptance of many OSS products implies that some effective mechanisms are being employed to overcome these barriers, though it remains largely unknown as to what these mechanisms are (Crowston et al., 2004; Gutwin et al., 2004; Scacchi, 2001). A few studies have shed light on this issue.

Using the case study methodology, Krogh et al. (2003) analyzed communication and coordination behaviors of the developers in the OSS community. The case that the researchers focused on is Freenet, a decentralized and anonymous peer-to-peer electronic file sharing network. The researchers conducted telephone interviews with thirteen developers and consulted project documents (e.g., the project's mailing-list, concurrent version system/subversion (CVS/SVN) repository, and FAQ). The major findings were: 1) a newly joined participant had to demonstrate his or her technical competency through submitting software code before gaining the writing-privilege of the CVS; 2) the developers specialized in coding certain modules of the project. This might be because of the contribution barriers associated with different modules in Freenet. The contribution barriers refer to a module's ease of modifying and coding, variation of computer languages, and modularity. The number of developers specialized in "easy" modules of Freenet was greater than the ones in "hard" modules. However, the study also found that a small number of developers were more "generalized" (i.e., contributing code to more than 13 modules) than "specialized" in order to better ensure the smooth integration of different modules.

Gutwin et al. (2004) conducted a case study to investigate how group awareness emerges in the OSS setting and helps the coordination among the members of an OSS team. The researchers interviewed fourteen developers from three highly successful OSS projects (NetBSD, Apache httpd, and Subversion). For additional information, they consulted the projects' mailing lists, chat archives, CVS/SVN repository, and documentations. Evidence shows that developers of an OSS project were able to form and maintain "a broad awareness of who are the main people working on their project, and what their areas of expertise are" (Gutwin et al., 2004, p. 75). Developers reached this broad awareness primarily through participating in mailing-lists, where the major design issues were discussed.

The study by Gutwin et al. is different from my research in two major aspects. First, my research utilizes TMS as its theoretical foundation. This theory has a wider scope than group awareness. As discussed in a subsequent section of this chapter and Chapter Three, the TMS is a multi-dimensional construct, encompassing knowledge differentiation, knowledge location, and knowledge credibility, whereas the group awareness in Gutwin et al.'s study only emphasized knowledge location. Second, this study adopts the survey method rather than the case study used by Gutwin et al.. The advantage of the survey method is the generalizability. Survey results might be generalized to the corresponding population through statistical inference. On the other hand, even though the case study is helpful in gaining deep understanding about a certain phenomenon, it is hard to generalize its findings to a wider population.

Kuk (2006) conducted a study to examine the communication and coordination pattern in OSS teams. He argued that participation inequality (i.e., the extent to which a small number of OSS participants dominate message postings and code contributions.) might be a common practice in many OSS projects. Participation inequality occurs because individual participants in collective action tend to interact strategically with only the most resourceful others in order to keep the cost of communication and coordination manageable. However, he also contended that concentrating interactions within too few participants certainly can lead to underutilizing the knowledge resources in groups. Therefore, the study hypothesized a curve-linear relation between participation inequality and knowledge sharing. Kuk tested the hypothesis by collecting the data from the K Desktop Environment (KDE, the graphical user interface for UNIX stations) developer mailing-list. The results show that the modest participation inequality exerted a positive impact on knowledge sharing behaviors in OSS teams, whereas even or extensive distribution of participation negatively affected knowledge sharing behaviors in OSS teams.

Crowston and Howison (2006) utilized the social network analysis technique to examine the extent to which communications in OSS teams are centralized within a few members and the degree to which communications are asymmetric, in other words, hierarchical. The researchers collected communication data of 174 OSS teams from three large OSS communities: Sourceforge, GUN Savannah, and Apache. The data demonstrated that the communications were highly hierarchical in general, but not uniformly centralized. However, this result must be interpreted with caution. The data were only collected from the debugging process; thus it is reasonable to question whether the pattern the data confirmed is generalizable to the entire development process.

OSS Team Performance

Researchers have studied different aspects of the OSS phenomenon to determine contributing factors to the OSS team performance. Gallivan (2001) claimed that OSS team effectiveness is dependent upon a set of control mechanisms, and thus proposed three control mechanisms: efficiency, calculability, and predictability. Team effectiveness was defined as a multiple-dimensional construct, and subsequently, assessed it from the quality of team output, the psychological needs of team members, and the cohesion of the team. Gallivan used the content-analysis technique to test the relations between the proposed control mechanisms and team effectiveness. Results showed no support for the hypothesized relations. This might be because other social and self-control mechanisms, such as individual reputation and membership management, rather than the proposed ones are more salient in the OSS settings (Gallivan, 2001).

Michlmayr (2005) attempted to verify the applicability of traditional software engineering practices in OSS projects. Specifically, he looked into the impacts of software process maturity on OSS project success. Software process maturity was assessed from the usage of collaboration tools (e.g., CVS and mailing lists), documentation, formal system testing (e.g., release plan, automatic testing suite, and defect tracking system), and software portability. Michlmayr evaluated project success based on the number of user downloads. Data was collected from 80 OSS projects hosted on SourceForge.net, one of the largest hosting sites for OSS projects. Results indicated that successful OSS projects were more likely to use collaboration tools and have formal system testing.

Based on social capital theory, Grewal, Lilien, and Mallapragada (2006) hypothesized that network embeddedness influences OSS project success. They defined three subconstructs, structural, junctional, and positional embeddedness, to represent network embeddedness, and defined project success as technical achievement (measured by the number of CVS commits) and commercial success (measured by the number of downloads). The researchers collected the data from 108 OSS projects involving 490 developers to test the hypothesized relations. Overall, the data showed that network embeddedness had significant impacts on the OSS projects' technical achievement and commercial success. Yet the directions of the impacts were somewhat equivocal and needed further investigation. In addition, no statistically significant association surfaced between technical achievement and commercial success of the OSS projects.

Next, Stewart and Gosain (2006) postulated that the OSS ideology can affect an OSS team's effectiveness. Therefore, they carried out a field study to explicate the content of the OSS ideology and the mechanism through which the ideology influences team effectiveness. Ideology refers to "shared, relatively coherently interrelated sets of emotionally charged beliefs, values, and norms that bind some people together and help them make sense of their worlds" (Trice & Beyer, 1993, p. 33). Stewart and Gosain (2006) defined team effectiveness from two aspects: input effectiveness (operationalized as team size and team effort) and output effectiveness (operationalized as task completion). The study surveyed 67 OSS team administrators. The results showed that two sets of OSS values (i.e., collaborative and individual values), three sets of OSS norms (i.e., distribution, forking, and named credit norms), and two sets of OSS beliefs (i.e., process and freedom beliefs) were salient in the OSS teams. These ideology components affected team effectiveness through communication quality, affective trust, and cognitive trust. The study also reported that task completion was positively related to team effort, but not team size.

Summary of OSS Research

An impressive body of research addresses a variety of questions regarding the governance and organization of the OSS development process. First, OSS teams, especially the successful ones, consist of three layers of participants: a small number of core developers, a large number of co-developers, and an even larger number of beta-testers. Second, researchers have found that communication and code contributions are often concentrated on few members of an OSS team; moreover, the existence of group awareness helps members' coordination. Finally, research has shown that OSS team performance is attributable to many factors, including social control mechanisms, software process maturity, network embeddedness, and the OSS ideology.

Surprisingly, no studies have examined how the developers overcome the daunting coordination barriers in the OSS setting to coordinate their knowledge even though knowledge coordination is the most important activity for a software development team (Robillard, 1999;

Rus & Lindvall, 2002; Tiwana, 2004). Therefore, this dissertation intends to fill the gap by applying the knowledge management literature and the theory of TMS to explicating knowledge coordination in OSS project teams.

Knowledge Management Research in the Context of Software Development Teams

Today's organizations rely heavily on teams (Cummings, 2004) to take advantage of valuable knowledge resource throughout the organizations for important organizational tasks, such as new product design, operation management, and information systems development. However, the potential of teams is not fully realized if the team members seldom share and integrate their unique task-relevant knowledge (Nonaka & Takeuchi, 1995). Therefore, a growing interest in understanding knowledge process in teams has developed (Austin, 2003). Knowledge process is especially critical for teams developing software because of its knowledge intensive nature (Robillard, 1999; Rus & Lindvall, 2002; Tiwana, 2004). Therefore, knowledge management research conducted in the context of the software development teams is reviewed in the following.

Knowledge Management and Performance of Software Development Teams

Faraj and Sproull (2000) asserted that the expertise, defined as "the specialized skills and knowledge that an individual brings to the team's task" (p. 1555), is the most important resource for a software development team. However, without purposely sharing and integrating the expertise, its mere presence cannot guarantee the production of high quality software. In other words, the expertise within a software development team must be coordinated. Therefore, Faraj and Sproull proposed the concept of expertise coordination, and conceptualized it as a three-dimensional construct which encompasses knowing expertise location, recognizing the need for expertise, bringing expertise to bear. The researchers hypothesized that expertise coordination should positively affect team performance; furthermore, the effects should be more salient than those of conventional team factors, such as presence of expertise, professional experience, administrative coordination, and software development methods.

The study surveyed 78 software development teams in a large U.S. high-tech firm. Of these, 69 teams completed the surveys. These teams were developing business application software, and each team had fewer than 15 full-time members. Project shareholders rated each team's performance in terms of effectiveness and efficiency.

The empirical data gathered generally supported the proposed hypotheses. Specifically, results showed that knowing expertise location was the most influential dimension of expert coordination. It positively affected both team effectiveness and efficiency. On the other hand, recognizing the need for expertise influenced only team effectiveness, and bringing expertise to bear affected only team efficiency. In addition, the effects of expertise coordination on team performance were more pronounced than those of conventional team factors (i.e., presence of expertise, professional experience, administrative coordination, and software development methods).

Subsequent to Faraj and Sproull's research on the concept of expertise coordination, Tiwana (2004) examined knowledge integration in the software development process. Particularly, he investigated how and to what extent integrating technical and application domain knowledge influences software development. Software development teams from Russia, Ireland, and India participated in the study. Results demonstrated that knowledge integration had significant influence on team performance: knowledge integration had positive relations with customer satisfaction and budget utilization, and a negative relation with the number of bugs in software developed.

Later, Mitchell (2006) examined knowledge management in Enterprise Application Integration (EAI) projects. Specifically, she studied how management's ability to access external knowledge and integrate internal knowledge affects project completion time. The study included 75 health networks. Each consisted of 20 or more health care facilities such as hospitals, specialty institutions, and physical groups. Since health networks adopted the two-tier structure to manage their EAI projects, CIOs at the corporate-level and IT managers at facility-level were surveyed. The researcher's primary findings are that (1) the access to external knowledge by both CIOs and IT managers was important for on-time project completion; (2) while the ability of the CIOs to integrate internal knowledge was critical for on-time project completion, this same ability did not matter for IT managers.

Antecedents of Knowledge Management in Colocated Software Development Teams

Even though the positive relation between knowledge management and software development team performance has been established, little is known about what determines the effectiveness of knowledge management in software development teams (Joshi & Sarker, 2006; Joshi, Sarker, & Sarker, 2004, 2005). Therefore, Joshi et al. designed a series of studies to search for the salient factors influencing knowledge transfer behaviors in colocated software development teams. They defined knowledge transfer as the extent to which one member has learned and internalized knowledge originating from other members of the same team.

Student teams participated in the studies. Each team developed an application system for a real client company. The studies consistently revealed the importance of communication and knowledge source credibility for knowledge transfer in colocated software development teams. That is, an individual member is more likely to internalize knowledge from other members who are highly communicative and credible. Meanwhile, the member himself or herself must have sufficient absorptive capacity and be motivated to learn in order to be an effective knowledge recipient.

Ojha (2005), on the other hand, focused on the influence of members' demography on knowledge sharing behaviors. Based on the similarity-attraction paradigm, he proposed that knowledge sharing is more likely to occur between team members with similar demographic characteristics because they are inclined to conduct effective socialization. To test the hypothesis, the researcher surveyed 588 software developers in 26 different Indian organizations. The results indicated that dissimilarity in martial status, gender, and education indeed had negative impacts on knowledge sharing among the members of a team. Another finding is that the members with longer organizational tenure were less likely to share their knowledge with others. Ojha reasoned that because these members had already accumulated sufficient knowledge to handle their tasks, they were less motivated to seek additional knowledge from others.

Ko, Kirsch, and King (2005) looked into how external consultants transfer ERP implementation, operational, maintenance, and training knowledge to ERP implementing organizations. The researchers conceptualized knowledge transfer as "the communication of knowledge from a source (e.g., a consultant) so that it is learned and applied by a recipient (e.g., an organization user)" (Ko et al., 2005, p. 62). Drawn from general knowledge transfer literature, three classes of factors were posited to influence knowledge transfer: knowledge related, communication-related, and motivational factors.

The study by Ko et al. (2005) surveyed 96 projects from 80 ERP implementing organizations and 38 consulting firms. The results showed that knowledge-related factors and motivational factors exerted significant direct effects on knowledge transfer, whereas the impacts of communication-related factors were mostly indirect. Specifically, (1) among knowledge-related factors, the recipient's absorptive capacity and shared understanding between a source and recipient facilitated knowledge transfer, but an arduous relationship between a source and recipient was detrimental to knowledge transfer; (2) among communication factors, the source's encoding competence and the recipient's decoding competence indirectly helped knowledge transfer. In addition, source credibility affected knowledge transfer directly as well as indirectly through the arduous relationship; (3) among motivational factors, only intrinsic motivation was significant for transferring ERP related knowledge.

He, Butler, and King (2007) maintain that empirical studies about software team's cognition are not sufficient even though it has been theorized to play an important role in effective knowledge utilization within software teams. Thus, these researchers designed a study to examine the formation process of team cognition, and hypothesized that team cognition has two dimensions: awareness of expertise location and shared task understanding. Participants in the study were 51 undergraduate database development teams. The empirical results showed that intra-team interactions and the team characteristics had significant influences on team cognition. More specifically, face-to-face meetings and phone calls helped shape team cognition; however, e-mail had no effect. Another finding is that gender diversity and familiarity among team members positively influence team cognition. In addition, team cognition evolved over time.

Knowledge Management in Distributed Software Development Teams

Because of the innovations in information and communication technologies (ICTs), the shortage of skilled software developers, and the pressure for cutting the software development cost, software development teams have increasingly adopted the distribution mode as the new organizational form (Carmel & Agarwal, 2002). Nevertheless, using the distributed mode brings new challenges, such as temporal, geographical, and cultural differences, to the already complex software development process (Damian, Lanubile, & Oppenheimer, 2003). As a consequence, research needs to be conducted to identify the determinants of knowledge transfer in this new context (Sarker, Sarker, Nicholson, & Joshi, 2005).

Answering this call, Kotlarsky and Oshri (2005) carried out an ethnographic study on two global software development (GSD) teams to examine whether team members are able to overcome differences in geographic distance, time zones, and culture to establish social ties and effectively share knowledge. The researchers also looked at how social ties and knowledge sharing affect collaboration. Results demonstrated the operation of social ties and knowledge sharing in both GSD teams and the positive impacts of these constructs on successful collaboration. Moreover, the evidence suggested that social ties and knowledge sharing were as important as collaborative tools (e.g., groupware, video-conferencing, and intranet) to successful collaboration.

Another study by Sarker et al. (2005), examined four specific factors influencing knowledge transfer behaviors in virtual software development teams: capability difference (e.g., technical know-how and teamwork skills), cultural diversity among the team members, credibility of the knowledge source, and communication volume. The study involved 12 virtual software development teams formed from 96 university students. The task was to develop application systems for some real business client companies. The communication among team members mainly relied on ICTs, such as online chat, document sharing, and threaded discussion. Results showed that communication volume was the strongest predictor of knowledge transfer. However, the researchers measured communication volume by asking each participant to "specify the extent of communication he or she had had with each remote team members" (p.

208) rather than counting the number of message exchanged. Therefore, the data suggests that the quality of communication might play a significant role in knowledge transfer since meaningless communication has little impression on people.

Sarker et al. (2005) also found that credibility of knowledge source and collectivism culture were positively associated with knowledge transfer. Furthermore, credibility strengthened the relation between communication and knowledge transfer. Surprisingly, capability difference did not affect knowledge transfer at all. Based on these findings, the authors suggested that highly communicative individuals with collectivism culture background are the ideal candidates to work in virtual software development teams. These individuals are more likely to discover what other team members know and share their own expertise with others. Doing so, in turn, creates new knowledge.

Summary of Knowledge Management Research in the Context of Software Development Teams

In spite of the importance of knowledge management in software development teams, the relevant research is scant. Only a handful of related studies have been found in the current literature. Nevertheless, these studies provide some useful theoretical insights and managerial directions for software development. First, the research has demonstrated that knowledge sharing and coordination improve the performance of software development teams, for example, producing higher quality software, improving the customer satisfaction, and reducing the development cost. Second, studies have identified three categories of factors as affecting knowledge management in software development teams: 1) characteristics of the knowledge source (e.g., credibility); 2) characteristics of the knowledge recipient (e.g., ability to absorb transferred knowledge); 3) communication behaviors between the source and recipient (e.g., communication volume).

TMS Research

Experimental Studies of TMS

Early research on TMS was mainly confined to experimental settings. The results generally support the hypothesis of TMS and report the positive relation between group performance and TMS. For example, Wegner et al. (1991) designed an experiment to indirectly demonstrate the existence of TMS, using 59 heterosexual dyads. The dyads were either dating couples or strangers paired for the purpose of the experiment. Each dyad needed to remember a number of words embedded in sentences without communication. While the researchers assigned some dyads with a clear memory structure about who should remember what, other dyads did not receive such instruction.

Wegner et al. hypothesized that, without assigned memory structure, the dating couples would remember more words than the stranger pairs because the dating couples have developed TMS. TMS has a memory structure which implicitly allocates the memory tasks (e.g., who should remember what) based on each member's domains of expertise. On the other hand, stranger pairs do not have developed TMS, and thus one member in a pair might try to remember the same words that the other member has already remembered. Apparently, this is not the optimal way to utilize memory space. This ineffective memory utilization, in turn, reduces the total number of unique words that the stranger pair can remember. The results of the experiment supported the hypothesis: Dating couples remembered more words (Mean = 31.40) than stranger pairs (Mean = 27.64), with $t(55) = 1.69$ and $p < 0.05$. This indirectly demonstrates the existence of TMS in the dating couples.

Wegner et al. also predicted that the dating couples would remember fewer words than the non dating pairs if the experimenter explicitly told each couple who should remember which words (i.e., an explicit memory structure) without considering the TMS. The rationale behind this hypothesis is that the explicit structure might interfere with the memory structure implied by the dating couple's TMS (i.e., the implicit memory structure). The results showed that the dating couples remembered fewer words (Mean = 23.75) than the stranger pairs (Mean = 30.14) in the presence of the explicit memory structure, with $t(55) = 2.90$ and $p < 0.005$, and hence supported the hypothesis.

Following the paradigm of Wegner's experiment discussed above, Hollingshead (1998a) went one step further to study the formation and operation of TMS. Specifically, she looked into how communication affects these two stages of TMS. For the experiment, dating couples or non dating pairs formed 88 dyads. The task for each dyad was to remember 36 words together. The researcher allowed some dyads but not all to communicate during the task.

Results revealed an interesting pattern of communication effects on TMS. When the researcher allowed communication, the members of non dating pairs, by communicating the information about each other's domains of expertise, were able to form a TMS. The TMS, in turn, better enabled the non dating pairs to remember the words. On the other hand, the dating couples had a tendency to use communication to devise some new memory structure to remember the words. The new structure, however, interfered with the memory structure implied by their existing TMS. As a consequence, the dating couples remembered fewer words than the strangers.

Nevertheless, when the researcher did not allow communication at all, the results were reversed: the dating couples tended to use the implied memory structure and hence remembered more words than the strangers. This experiment points out that communication might help the formation of a TMS, but unrestricted communication might impede the proper function of the TMS once it is formed.

Hollingshead (1998b) later conducted two more experiments focusing on the effects of communication and its modes on the retrieval process in TMS. The experiments again involved dating couples and stranger pairs. Each couple performed a task that demanded the couple to pool together their individually-held knowledge. Some couples communicated face-to-face, while others communicated via a computer conferencing system.

The results indicated that the dating couples in the face-to-face condition performed better than the dating couples in the computer-mediated condition. Additionally, the dating couples in the face-to-face condition performed better than the stranger pairs in both face-to-face and computer-mediated conditions. However, Hollingshead could not directly attribute better performance to knowledge about the distribution of expertise within a couple. Instead, better performance was positively related to a transactive retrieval process. In this process, a reference one member of a couple makes to some aspect of an event reminds the other's memory about other aspects of the same event. As a result, the couple together is able to retrieve more information about the event than an individual member. Additionally, the results showed the frequency of transactive retrieval was much higher in face-to-face dating couples than the others. Therefore, Hollingshead concluded that the transactive retrieval mediates the effects of TMS on performance.

These two experiments validate Wegner's early conceptualization about the importance of transactive integration in TMS. In addition, they suggest that computer-mediated communication may not be an optimal communication mode for retrieving knowledge stored in

TMS because certain non-verbal cues, such as eye contact and voice tone, are not available (Hollingshead, 1998b). However, it is conceivable that the couples might adapt their communication to such a lean medium through repeated experiences (Hollingshead, 1998b).

The experiments Wegner et al. and Hollingshead conducted provide some preliminary insight about the nature of TMS, yet they carry two major limitations. First, these experiments studied only TMS of the dyads. But Wegner et al. (1985) postulate that TMS also operates in larger groups than the dyads, such as teams, organizations, and even societies. Therefore, experiments examining TMS in larger groups, such as teams, were needed. Second, the validity of TMS thus far was demonstrated only through indirect evidence (e.g., difference in the number of words remembered between dating couples and stranger pairs), and no researchers had directly measured TMS. Hence, a method to measure TMS directly was necessary in order to advance the TMS research.

To fill the gap, Moreland and his colleagues (Moreland, 1999; Moreland et al., 1996) designed and conducted a series of laboratory experiments to examine the formation of TMS within a self-managed group and to identify the impacts that TMS has on group performance. They argued that shared experience of group members (e.g., working together on group tasks) leads to the gradual development of a TMS. This TMS improves the group performance because knowing who is good at what helps the group better utilize its human resources (e.g., matching the task with members' expertise) (Moreland, 1999; Moreland et al., 1996). Moreover, the TMS ensures the effective coordination among the group members since they now can forecast each other's behaviors and plan their own behaviors accordingly (Moreland, 1999; Moreland et al., 1996).

Moreland et al. (1996) recruited undergraduates for the experiments, and trained them either in groups or individually to assemble the AM portion of a radio. The researchers videotaped each group as it assembled the radio. Two judges later watched the video to evaluate three behaviors of each group: memory differentiation (i.e., "the tendency for group members to specialize in remembering different aspect of the assembly process" (Moreland, 1999, p. 10)), task coordination (i.e., "the ability of group members to work together efficiently on the radio" (Moreland, 1999, p. 10)), and task credibility (i.e., "the level of trust among group members in one another's radio knowledge" (Moreland, 1999, p. 10)). Moreland et al. operationalized these three behaviors as the manifestations of TMS.

Results revealed that the groups with the members trained together scored higher on the TMS and performed better than the groups with the members trained apart. Further analysis uncovered that the TMS indeed mediated the relation between group training and performance. The experiments also found an alternative way to nurture the TMS: simply informing everyone in the group about the expertise of the other members. Results demonstrated that this method had similar effects on the group performance as training group members together.

The TMS Scale

A common concern with the experimental studies is the external validity because of artificiality of the experimental conditions (Crano & Brewer, 2002). Therefore, Lewis (2003) expended significant effort to develop and validate a field TMS scale, which facilitates the TMS theory testing in organizational settings. According to the theoretical definition (Wegner, 1987; Wegner et al., 1985) and previous experimental research (Moreland, 1999; Moreland et al., 1996), Lewis hypothesized the TMS as a multi-dimensional construct. She proposed three dimensions to "tap the essence of the TMS construct" (Lewis, 2003, p. 590). These dimensions are specialization (i.e., the extent to which the group members' domains of expertise differentiate), credibility (i.e., the extent to which the group members have confidence in each other's expertise), and coordination (i.e., the extent to which the group members integrate their different domains of expertise to accomplish group tasks). Lewis created 5 items to measure each dimension, resulting in a 15-item scale. The scale was first tested in a laboratory experiment and then in two field studies. Both the experiment and field studies confirmed the three dimensions of the TMS construct and consistently demonstrated the internal consistency and construct validity of the TMS scale.

Using her newly developed scale, Lewis (2004) researched how a TMS emerges and then evolves into a mature state. Teams, comprised of MBA students, participated in the study. Each team completed a semester-long consulting project. The project had two phases: planning and implementation. Results show that a TMS initially emerges at the end of the planning phase and continues to evolve during the implementation phase. The initial condition of a team (e.g., the distribution of expertise and prior familiarity among the team members) and communication frequency among the team members determines the TMS emergence. Specifically, a TMS is more likely to emerge in a team with members who have diversified expertise domains, are familiar with each other, and frequently communicate face-to-face. But non-face-to-face communication (e.g., email and telephone calls) in the planning phase has no effect on the initial TMS.

Further, Lewis (2004) found that the initial TMS continues to evolve in the implementation phase. The evolution is still dependent on the communication frequency. However, the role of non-face-to-face communication in the TMS's evolution is somehow more complicated. When the initial TMS is not well-established, more frequent non-face-to-face communication, in addition to face-to-face communication, produces the more mature implementation-phase TMS. On the other hand, if the initial TMS is well-established, too much non-face-to-face communication is detrimental to the further maturity of the TMS.

Field Studies of TMS

Even though ample evidence demonstrating the positive relation between TMS and group performance in "artificial" groups (e.g., the group formed for the purpose of studying TMS), such a relation had not yet been sufficiently examined in natural organizational groups (Austin, 2003). Consequently, Austin (2003) conducted a field study to investigate the relation between TMS and group performance in natural organizational setting. He posited that that each group should have two TMS: one for group task and the other for external relationship (e.g., the relationships between group members and nongroup members). Furthermore, he proposed four dimensions for each TMS posited: group knowledge stock, transactive memory consensus, knowledge specialization, and transactive memory accuracy.

The empirical data shows that the relations between each individual TMS dimension and group performance are quite complex. Among four dimensions of task TMS, knowledge specialization positively affects external and internal evaluation of group performance, and transactive memory accuracy positively influences goal attainment, external, and internal evaluation of group performance. Nevertheless, external relationship TMS appears unimportant to group performance at all.

Akgün et al. (2006) carried out another field study to analyze the effects of TMS on process effectiveness and product success of new product development teams. However, the researchers operationalized the TMS as a unidimensional construct in this study, and defined it as the shared awareness of knowledge distribution among the members of a new product development team. Teams from a variety of industries, such as telecommunications, software, and machinery, participated in the study.

The results show that TMS positively influences the team process. The improved team process, in turn, produces successful new product development. Moreover, the effects of TMS on team process are partially mediated by the collective mind, which manifests as heedful interactions among the team members. These findings suggest that merely knowing the location of knowledge might not lead to full usage of the knowledge. Collective mind, on the other hand, helps integrate the distributed knowledge within new product development teams. In addition, the relation between TMS and team process is subject to the environmental constraints. In the study, when the environment was highly uncertain about technological development and customer needs, the relation became negative due to the invalid TMS. From these results, one may infer that an effective TMS should be updated as the surroundings of a team change (Akgün et al., 2006).

The TMS in Virtual Teams

Yoo and Kanawattanachai (2001) studied the development of TMS in virtual teams. Subjects of the study were 38 virtual teams of MBA students. The task for each team was to manage a hypothetical high-tech company for eight weeks. Each team's members communicated through an online discussion forum and an e-mail listing. The results revealed that communication volume was essential for the formation of TMS. The participants reported that they explicitly communicated the information about each other's expertise in the early phase of the task to decide what role each member should assume in the task. High communication volume thus helps to nurture TMS, which, in turn, improves team performance.

In a follow-up study, Kanawattanachai and Yoo (2007) specifically examined three behavioral dimensions of the TMS in virtual teams, including expertise location, cognition-based trust, and task-knowledge coordination. The study found that task-oriented communication in the early phase of team project helped locate the expertise in the team and establish cognition-based trust among the team members. However, these positive effects diminished later on. Expertise location and cognition-based trust consistently promoted task-knowledge coordination over time. But the task-knowledge coordination began to exert positive influence on team performance only when it was fully matured. Moreover, task-orientation communication had positive effects on team performance, but the effects diminished with the maturity of the TMS.

Other TMS Studies

Besides experimental and field studies, researchers adopted other methodologies to study TMS. For instance, Palazzolo, Serb, She, Su, and Contractor (2006) used the computational modeling technique to explore how the initial conditions of a team influence the development of its TMS and what role communication plays in the process. Three initial conditions particularly considered in the study were initial knowledge level of the team members, initial accuracy of expertise recognition among the team members, and team size. The TMS was defined in terms of "the extent to which the TM system has developed or matured: (a) accuracy of expertise recognition and (b) knowledge differentiation" (Palazzolo et al., 2006, p. 226).

Results demonstrated the central role of communication in the development of TMS. Frequent communication increased accuracy of expertise recognition and promoted the knowledge differentiation among the members of a team. Furthermore, the initial conditions of a team and team size had significant impacts on its TMS, but the impacts were largely mediated through communication: a small team with the members having low knowledge level have strong potential to form an effective TMS through communication.

Summary of TMS Research

A large number of experimental and field studies have demonstrated the importance of the TMS for group performance (Akgün et al., 2006; Austin, 2003; Hollingshead, 1998b; Lewis, 2004; Wegner et al., 1991). Therefore, the researchers have been searching the antecedents of the TMS. Communication turns out to be the most salient determinant (Hollingshead, 1998a, 1998b; Lewis, 2004; Palazzolo et al., 2006; Yoo & Kanawattanachai, 2001). Especially in the early stage of TMS development, a large communication volume is essential (Palazzolo et al., 2006; Yoo & Kanawattanachai, 2001). During this early development of the TMS, the members of a newly formed group explicitly exchange a large amount of background information about one another, such as education level, work experience, and professional association (Yoo & Kanawattanachai, 2001). Such information subsequently serves as a starting point for knowledge differentiation – the initiating phase of the TMS (Wegner, 1987). However, when the TMS is fully developed, communication quality, rather than communication volumes, appears to be important. Actually, excessive communication might be detrimental to the relation between the TMS and group performance, for unnecessary communication interferes with the operation of the TMS (Hollingshead, 1998a, 1998b). Team size has also been found to affect the TMS formation (Palazzolo et al., 2006), but the effect is negative because it is increasingly hard to identify the expertise in the team as the size of the team increases (Palazzolo et al., 2006). Additionally, through a series of laboratory studies, Moreland (1999) concluded that shared experience facilitates the TMS formation.

Despite rich TMS studies, two particular areas of TMS call for further research: First, the structure of the TMS construct is unclear. Even though most TMS researchers (e.g., Austin, 2003; Lewis, 2003, 2004; Moreland, 1999; Palazzolo et al., 2006; Kanawattanachai & Yoo, 2007) agree that the TMS is a multi-dimensional construct, there is no consensus on what the TMS's dimensions are. For example, Moreland et al. (1999) and Lewis (2003, 2004) posit three dimensions of the TMS (i.e., specialization, credibility, and coordination), whereas Austin (2003) hypothesizes four dimensions of the TMS (i.e., group knowledge stock, transactive memory consensus, knowledge specialization, and transactive memory accuracy). Yet, Palazzolo et al., (2006) propose two dimensions of the TMS (i.e., accuracy of expertise recognition and knowledge differentiation). Therefore, more research needs to be conducted to look into the structure of the TMS construct.

Second, research on the TMS in virtual teams is scant. Only a handful of studies are identified in current literature. Furthermore, the results of these studies are somewhat contradictory to each other. For example, Lewis (2004) reported that face-to-face communication is paramount for the TMS development, whereas non-face-to-face communication (e.g., emails) has no effect on the TMS development. Consequently, she concludes that virtual teams might not be able to develop the effective TMS. Kanawattanachai and Yoo (2007), on the other hand, found that virtual teams indeed form the TMS mainly through frequent communication in mailing-lists. The virtual TMS, in turn, improves team performance. Therefore, more research needs to be conducted to resolve the inconsistent findings about the TMS development in the virtual context.

This dissertation aims to address the above gaps in the literature. Specifically, this study hypothesizes that the TMS is a three-dimension construct (i.e., knowledge differentiation, knowledge location, and knowledge credibility). These three dimensions of TMS positively influence OSS team performance through facilitating knowledge coordination and improving the quality of communication among members of an OSS teams. Furthermore, this study argues that frequent communication via online coordination tools (e.g., mail-listing and CVS) and team size are salient antecedents for TMS in an OSS team. The details of the hypotheses are discussed in the following sections.

CHAPTER THREE: RESEARCH HYPOTHESES

This chapter presents the hypotheses of this research. These hypotheses are developed based on the literature reviewed in Chapter Two. First, the dimensions of the TMS are discussed. Second, antecedents of the TMS formation are proposed. Third, the relations between the TMS, knowledge coordination, and communication quality are hypothesized. Lastly, the relations between knowledge coordination, communication quality, and team performance are discussed.

Transactive Memory System

As prior research has demonstrated (Austin, 2003; Lewis, 2003, 2004; Moreland, 1999; Palazzolo et al., 2006), TMS is a multi-dimensional construct. Therefore, this study particularly considers three aspects of the TMS: knowledge differentiation, knowledge location, and knowledge credibility.

First, this study defines knowledge differentiation as the extent to which the developers of an OSS team specialize in different knowledge domains. Wegner (1987; see also Wegner et al., 1985) claims that TMS is essentially a knowledge-holding structure where diverse domains of knowledge from different team members are stored and indexed. This knowledge-holding structure starts to form as team members accept responsibility for knowledge of different domains that are relevant for team tasks (Wegner, 1987; Wegner et al., 1985). This responsibility allows each team member to develop a distinct and non-redundant knowledge specialty (Lewis, 2003), rather than reproducing knowledge that other team members already possess (Palazzolo et al., 2006). Consequently, the team, as a whole, has a comprehensive knowledge base to draw on. Knowledge differentiation is especially important for a software development team (Faraj & Sproull, 2000) because software development typically involves integrating knowledge from many different domains, such as software architecture, software design methodologies, and users' business application domain knowledge (Rus & Lindvall, 2002; Tiwana, 2004). "Having differentiated expertise facilitates the sharing of the tasks and allow goals to be accomplished collectively" (Palazzolo et al., 2006, p. 226). In addition, "a differentiation knowledge base reduces the workload for team members because they can simply go to an expert for information" (Palazzolo et al., 2006, p. 226). Therefore, this study considers knowledge differentiation as one of the TMS dimensions.

Second, as Wegner (1987; see also Wegner et al. 1985) originally conceived, shared understanding among team members about who knows what (i.e., knowledge location) is the central element of TMS. This location information functions as an important integrative mechanism to coordinate behaviors between team members (Faraj & Sproull, 2000). Recent research shows that knowledge location is especially critical for software development team performance (He et al., 2007). Therefore, this study defines knowledge location as the extent to which the developers of an OSS team are familiar with the distribution of task relevant knowledge within the team.

Finally, past studies suggest that knowledge credibility is another dimension of the TMS (Lewis, 2003; Moreland, 1999). For instance, in an experimental study, Moreland (1999) observed that team members not only need to know each other's expertise but also must have sufficient trust in each other's expertise. This trust makes team members willingly to internalize knowledge from others (Joshi & Sarker, 2006; Joshi et al., 2004, 2005), allows team members to carry out tasks of their specialties without explicitly justifying their course of action (Liang, Moreland, & Argote, 1995), and avoid criticizing each other's work too often (Moreland, 1999). Drawing on Moreland's study, Lewis (2003) also posited knowledge credibility as one dimension of the TMS, and conducted one experiment and two field studies to test this proposition. The results from the studies all support her hypothesis about knowledge credibility. Therefore, this study conceptualizes knowledge credibility as one dimension of the TMS, and defines knowledge credibility as the extent to which the developers of an OSS team have confidence in each other's knowledge.

Although several TMS researchers (e.g., Lewis, 2003; Moreland, 1999) suggest knowledge coordination as another dimension of the TMS, according to Wegner's conceptualization of the TMS (1987; see also Wegner et al., 1985), effective and efficient knowledge coordination takes place only after the TMS has been fully developed. Specifically, Wegner (1987; see also Wegner et al., 1985) theorizes that knowledge coordination is the mediating process through which the fully developed TMS affects group performance: When the TMS matures, knowledge integration ensues. The group, with the TMS pointing to who knows what, is capable of integrating the unique knowledge from different team members to produce solutions for the group tasks that are unsolvable if only relying on one member's knowledge. The recent study from Kanawattanachai and Yoo (2007) substantiated the mediating effect of knowledge coordination on the relation between the TMS and team performance. They found that expertise location and knowledge credibility consistently influenced knowledge coordination behaviors throughout different phases of a software project; however, knowledge coordination affected the team performance only at the final stage of the project, when the TMS was fully developed. Therefore, this study posits knowledge coordination as the mediator of the relation between the TMS and the OSS team performance rather than a dimension of the TMS.

Communication Volume and TMS

Communication plays an important role in the TMS development (Wegner, 1987; Wegner et al., 1985). Through intensive communication, members negotiate the knowledge responsibilities (Lewis, 2004; Wegner, 1987; Wegner et al., 1985), exchange information about each other's domains of expertise (Hollingshead, 1998a; Yoo & Kanawattanachai, 2001), and assess the credibility of each other's expertise recognition (Lewis, 2004; Palazzolo et al., 2006). Communication in an OSS team takes place primarily in the developer-mailing list (Gutwin et al., 2004; Krogh et al., 2003) where important project issues such as, release schedule, software architecture, and critical bugs, are discussed (Krogh et al., 2003; Scacchi, 2001). Due to the distributed and voluntary nature of OSS developers, not all of them participate in every discussion. However, by reading the archived postings, members still can maintain a fairly good awareness about who is active on the project, what modules they are working on, and what they can contribute (Gutwin et al., 2004).

Besides the developer-mailing list, a type of indirect communication occurs through commit logs. OSS teams use version control tools (e.g., CVS) to manage their source code (Louridas, 2006). When a developer commits a piece of new code to the source file, the version control tools generate a commit log to record what has been changed, who made the change, and when the change was made (Grewal et al., 2006). "Most developers keep an eye on the commits to stay up to date on what is happening on the project, and to watch for changes that may affect their own work or plans" (Gutwin et al., 2004, p. 77). The total number of developer-mailing-list postings and commit logs are used as a proxy for communication volume in this research. It is hypothesized that:

H1: Communication volume is positively associated with the TMS development in an OSS team.

H1a: Communication volume is positively associated with knowledge differentiation in an OSS team.

H1b: Communication volume is positively associated with knowledge location in an OSS team.

H1c: Communication volume is positively associated with knowledge credibility in an OSS team.

Figure 1 summarizes the hypotheses discussed above. The path from communication volume to knowledge differentiation corresponds to H1a. The path from communication volume to knowledge location corresponds to H1b. The path from communication volume to knowledge credibility corresponds to H1c.

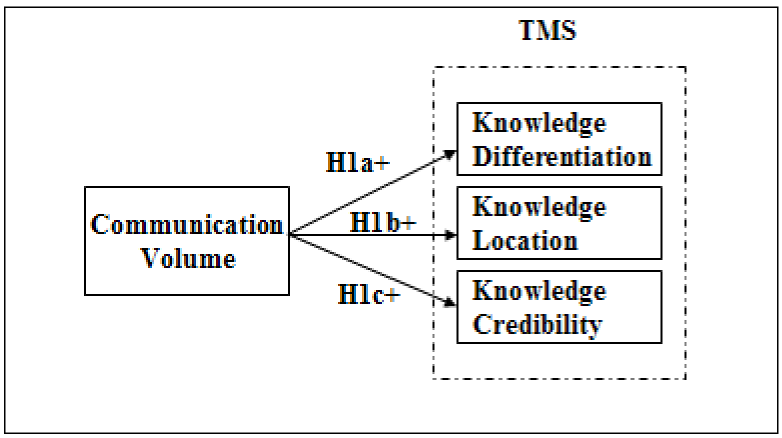


Figure 1. Hypothesis 1a, 1b, and 1c

Team Size and TMS

Due to the voluntary nature of the OSS workforce, OSS researchers suggest that the team with a large number of developers should have high potential to achieve its project goals because of a large knowledge resource on which to draw (Raymond, 2000; Stewart & Gosain, 2006). However, the TMS researchers argue for the opposite. Palazzolo et al. (2006) posit that it might be difficult for a large team to form a functional TMS because of its members' cognitive limitation. To be familiar with all others in the team, the members need to maintain information about the other members (Wittenbaum, Vaughan, & Stasser, 1998). As the size of the team increases, the amount of information each member needs to maintain builds up (Austin, 2003) and may eventually exceed the members' cognitive threshold (Palazzolo et al., 2006). As a result, the members in a large OSS team might have a hard time identifying the expertise of other members, the uniqueness of a member's expertise, and the quality of others' expertise. Therefore, it is hypothesized that

H2: The size of an OSS team is negatively associated with the formation of its TMS.

> H2 a: The size of an OSS team is negatively associated with knowledge differentiation in an OSS team.
>
> H2b: The size of an OSS team is negatively associated with knowledge location in an OSS team.
>
> H2c: The size of an OSS team is negatively associated with knowledge credibility in an OSS team.

Figure 2 summarizes the hypotheses discussed above. The path from team size to knowledge differentiation corresponds to H2a. The path from team size to knowledge location corresponds to H2b. The path from team size to knowledge credibility corresponds to H2c.

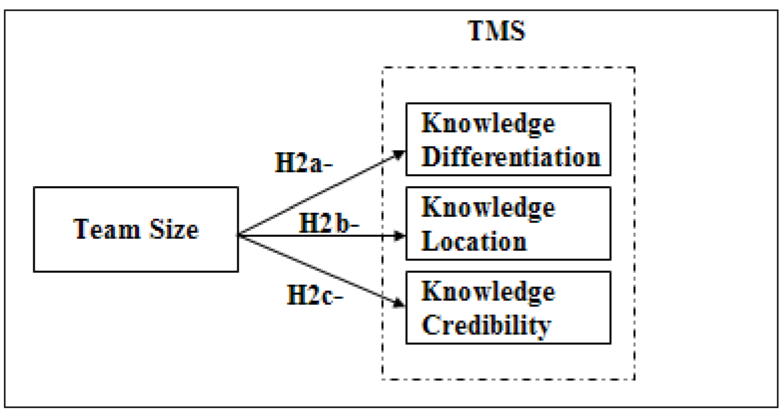


Figure 2. Hypothesis 2a, 2b, and 2c

Knowledge Coordination

Knowledge is the most important resource for a software development team (Faraj & Sproull, 2000; Rus & Lindvall, 2002; Tiwana, 2004), yet being fragmented among different members. Therefore, it must be heedfully coordinated to take effect on the team performance (Faraj & Sproull, 2000; Mitchell, 2006; Tiwana, 2004). Knowledge coordination in this research, adapted from knowledge management and TMS literature (e.g. Faraj & Sproull, 2000; Kotlarsky

& Oshri, 2005; Lewis, 2003; Moreland, 1999; Sarker et al., 2005), refers to the extent to which the members of an OSS team integrate their different domains of expertise on software development tasks. For example, the following extract from the developer-mailing list of the phpMyAdmin project demonstrates how two OSS developers pooled their relevant knowledge to solve a font size problem.

[name1 removed]:
> [name2 removed]:
…
>> I know that we have decided to use the default browser's font size, but
>> I just made a test:
>> [code omitted]…
>> and when I display that in FF 2.0.0.5, the text looks larger than the
>> same text as seen on the Operations panel from PMA 2.10.2.
>> In fact, I have to choose a font size of 80% to obtain the same size as
>> in my test.
>
> 'your' test is larger than in PMA and you have to reduce to 80% in PMA to
> get same size as in your test? how does this work?
Sorry, the text looks larger in PMA. In my test, it looks OK. It looks
too large in PMA, as mentionned by Florian on the users list.
>> Why? Is my simple test flawed?
> i cannot see any difference
Did you try my test page and compare it with PMA's output?
> did you tried adding html …..?
Sorry, I don't know where to add this. Modify the <html> tag?
> what tells you page information about view mode? standard compliance both?
If I do right-click/page information, it says "mode de respect strict
des standards" (strict standard mode).

Interactions such as the one above occur frequently in mailing lists: A developer posts a problem that he or she encounters when running the software, and then several other knowledgeable developers suggest and elaborate on a solution.

TMS and Knowledge Coordination

Previous research postulates that a fully developed TMS is beneficial for knowledge coordination because 1) knowledge differentiation allows a team to have a broad knowledge base on which to draw (Wegner, 1987; Wegner et al., 1985); 2) knowledge location directs knowledge seekers on where to find needed knowledge (Moreland, 1999; Wegner, 1987; Wegner et al.,

1985) and thus helps team learning (Akgün et al., 2006); 3) knowledge credibility makes a member more likely to accept the knowledge from others (Joshi & Sarker, 2006; Joshi et al., 2004, 2005). Therefore, it is hypothesized that

H3: The TMS is positively associated with knowledge coordination among the members of an OSS team.

> H3 a: Knowledge differentiation is positively associated with knowledge coordination among the members of an OSS team.
>
> H3b: Knowledge location is positively associated with knowledge coordination among the members of an OSS team.
>
> H3c: Knowledge credibility is positively associated with knowledge coordination among the members of an OSS team.

Figure 3 summarizes the hypotheses discussed above. The path from knowledge differentiation to knowledge coordination corresponds to H3a. The path from knowledge location to knowledge coordination corresponds to H3b. The path from knowledge credibility to knowledge coordination corresponds to H3c.

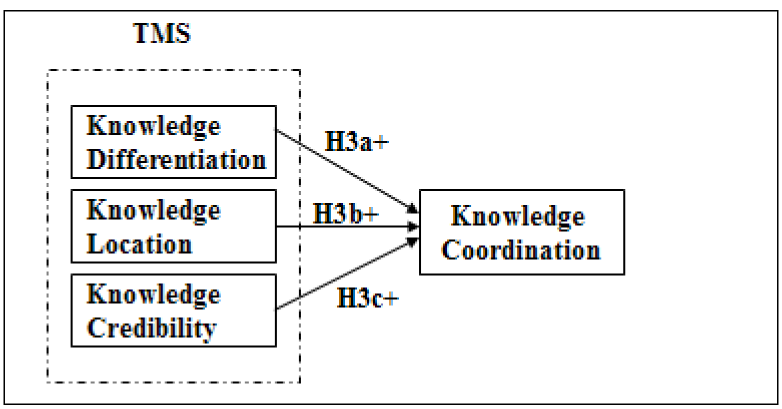


Figure 3. Hypothesis 3a, 3b, and 3c

Communication Quality

The timely and relevant communication among the members is important for an OSS team's outcomes (Stewart & Gosain, 2006). Therefore, this research defines communication quality as the extent to which communication is timely and relevant to OSS development tasks.

TMS and Communication Quality

A mature TMS can improve the quality of communication among members within an OSS team because knowledge differentiation enables the members to specialize in different expertise domains (Wegner, 1987; Wegner et al., 1985). This specialization ensures that detailed and most relevant knowledge is communicated when dealing with critical development tasks. Further, knowledge location and credibility inform the developers to whom they should talk to acquire needed knowledge. Therefore, it is hypothesized that

H4: The TMS is positively associated with communication quality among members of an OSS team.

H4a: Knowledge differentiation is positively associated with communication quality among members of an OSS team.

H4b: Knowledge location is positively associated with communication quality among members of an OSS team.

H4c: Knowledge credibility is positively associated with communication quality among members of an OSS team.

Figure 4 summarizes the hypotheses discussed above. The path from knowledge differentiation to communication quality corresponds to H4a. The path from knowledge location to communication quality corresponds to H4b. The path from knowledge credibility to communication quality corresponds to H4c.

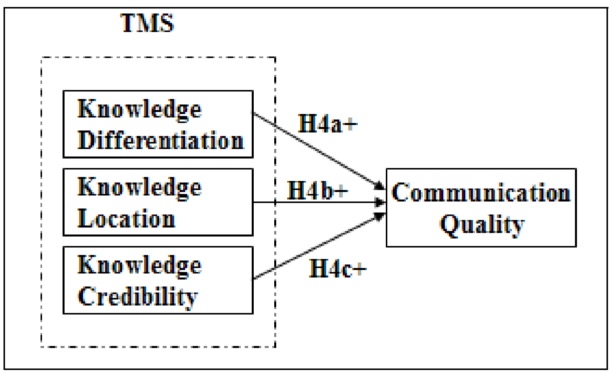


Figure 4. Hypothesis 4a, 4b, and 4c

Communication Quality and Knowledge Coordination

Through timely and relevant communication, the knowledge fragmented among different members of an OSS team is integrated on the development tasks. Therefore, it is hypothesized that

H5: The quality of communication among the members of an OSS team has positive effects on knowledge coordination in the team.

Team Performance

Grewal et al. (2006) suggest that the performance of an OSS team should be evaluated not only from the perspective of the technical achievement but also with regard to its commercial success. Technical achievement refers to the extent to which an OSS team has completed software development tasks (e.g., the percentage of bugs resolved) (Au, Carpenter, Chen, & Clark, 2009; Stewart & Gosain, 2006). Commercial success refers to the extent to which users have accepted the software that an OSS team has developed (e.g., the number of downloads) (Gallivan, 2001; Grewal et al., 2006; Michlmayr, 2005). This research agrees with this view. However, the focus of this study is the OSS team's internal coordination mechanism, which has direct bearing only on an OSS team's technical achievement, not on its commercial success. Although software's technical characteristics affect its user acceptance (Sabherwal, Jeyaraj, & Chowa, 2006), other market factors, such as the competitiveness of the market and the size of user population, might also play important roles in user acceptance, and thus confound the impacts of software's technical characteristics on user acceptance. Therefore, this study narrows its focus on the technical achievement of the team performance.

Knowledge Coordination and Technical Achievement

Several researchers have substantiated the positive relation between knowledge coordination and technical achievement in the software development team settings. Knowledge coordination is defined as to the extent to which the members of an OSS team integrate their different domains of expertise on software development tasks (e.g. Faraj & Sproull, 2000; Kotlarsky & Oshri, 2005; Lewis, 2003; Moreland, 1999; Sarker et al., 2005). For example, Faraj and Sproull (2000) found that expertise coordination improves the software development team's work quality. Tiwana (2004) also reported that knowledge coordination is positively associated with the reliability of software produced because teams with effective knowledge coordination need less planning ahead of time, incur fewer misunderstanding and confusion while performing, and the members cooperate smoothly with each other (Liang et al., 1995). Therefore, it is hypothesized that

H6: Knowledge coordination within an OSS team positively affects the team's technical achievement.

Communication Quality and Technical Achievement

Poor communication leads to a variety of negative effects on software development, such as incompatible sub-modules, redundant coding, and unfulfilled user requirements (Brooks, 1975; Carmel, 1999). On the other hand, timely and relevant communication enables the developers to make informed design decisions and amass necessary information to solve emerging problems. This type of communication is especially significant for the OSS teams (Stewart & Gosain, 2006) since their members rarely, if ever, meet face-to-face. Therefore, it is hypothesized that

H7: The quality of communication among the members of an OSS team positively affects the team's technical achievement.

Figure 5 summarizes H5, H6, and H7 discussed above. The path from communication quality to knowledge coordination corresponds to H5. The path from knowledge coordination to technical achievement corresponds to H6. The path from communication quality to technical achievement corresponds to H7.

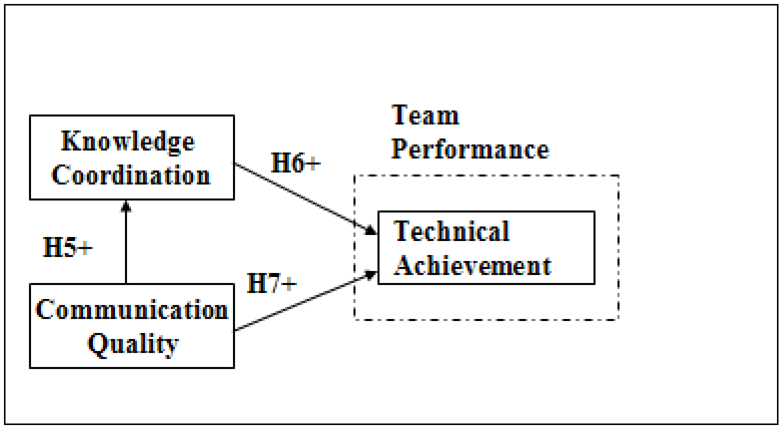


Figure 5. Hypothesis 5, 6, and 7

Figure 6 shows the research model, which summarizes the hypotheses developed above.

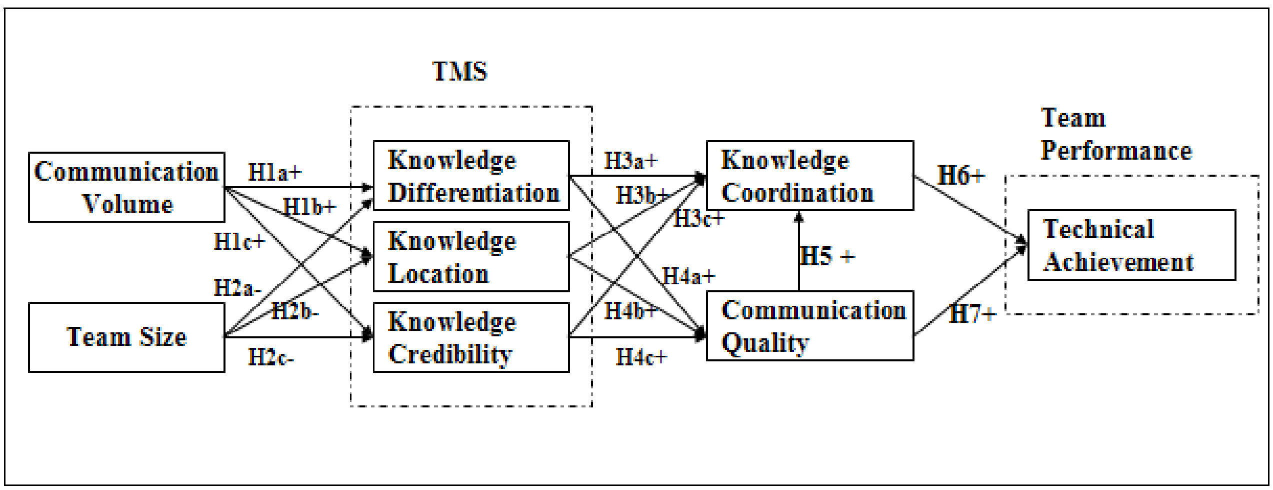


Figure 6. The Research Model

CHAPTER FOUR: RESEARCH METHODOLOGY

This chapter presents the methodology employed in this study. It discusses the research site and sample, forms of data to be collected, operationalization of the constructs of interest, and data collection procedures. At the end, it describes the statistical method to be used for data analysis and preliminary results of a pilot study.

Research Site and Sample

The research site for this study is Sourceforge.net. Sourceforge.net is one of the largest OSS online communities in the world. It hosts over 162,000 projects with a centralized resource for managing projects, communications, and code. To support collaborative development, Sourceforge.net provides each project with a multitude of tools, such as a project home page, CVS/SVN repository, and mailing-list. Projects on Sourceforge.net are broadly classified into fourteen categories: clustering, database, desktop, development, enterprise, financial, games, hardware, multimedia, networking, security, system administration, storage, and VoIP.

To limit the variance that might be introduced by project category differences (Stewart & Gosain, 2006), this study will draw its sample from two randomly selected project categories. Teams in this category must meet three criteria to be included in the study. First, since prior studies found that the TMS can be formed in dyads, the team included in this study must have at least two developers. Second, the project should have a history of at least six months. As Moreland et al. (1996; see also Moreland, 1999) have demonstrated in a series of experiments, it is the shared experience that allows a TMS to form and evolve. Third, to make sure that abandoned projects are excluded from the sample, the team must have been active (e.g., having new postings in the developer mailing-list or new commit logs in the code repository) for at least 60 days at the time of sample selection.

Overview of Data

The data will be gathered from two sources: survey answers from OSS project administrators and project event data from the project's website. A Web-based survey was developed for this research. The survey is structured so that respondents (i.e., OSS project administrators) first report the name of the OSS project with which they have been most involved, and subsequently assess knowledge differentiation, location, credibility, coordination, and communication quality of their corresponding teams using a series of survey scales. Most of the survey scales are adapted from prior studies, and have been validated through expert panels and a survey pilot. Appendix B contains the survey.

The second form of data is project event data. Sourceforge.net archives a large quantity of various OSS project event data, such as the date when the developer joins a team, the number of bug reported, and the number of postings in the mailing list. This data is publicly available from each project's website. This research will collect such data to operationalize several constructs of interest, including communication volume, shared task experience, team size, and technical achievement. The details of operationalizations are discussed in the section below.

Measurement and Construct Definitions

As previously discussed, two forms of data will be collected. This section describes how the data will be used to operationalize the constructs of interest. First, the survey items are explained, and then project event data are discussed.

Survey Items

Stone (1978) recommends using previously validated survey items to enhance the validity of measurement. Adopting this recommendation, initial items were first produced by largely adapting previously validated items. They were then reviewed by two expert panels and validated through field testing.

TMS Measures

Three dimensions of TMS are proposed: knowledge differentiation, knowledge location, and knowledge credibility. Based on Wegner's conceptualization of TMS (1987) and empirical work from other TMS researchers (e.g., Austin, 2003; Lewis, 2003, 2004; Moreland, 1999; Palazzolo et al., 2006), this research defines knowledge differentiation as the extent to which the developers of an OSS team specialize in different knowledge domains. Knowledge location is the extent to which the developers of an OSS team are familiar with the distribution of task relevant knowledge within the team. Knowledge credibility is the extent to which the developers of an OSS team have confidence in each other's knowledge. Using these definitions as the guideline, an extensive search was performed in prior TMS research to identify potential survey items that can be adapted to this study. Lewis (2003) had previously developed a five-item scale to measure the TMS specialization, consistent with knowledge differentiation in this research. She also developed another five-item scale to measure the TMS credibility, consistent with knowledge credibility in this research.

The two scales had been tested through one laboratory study and two field studies. The results demonstrated their internal consistency and validity. Thus, this study adapted these scales to measure knowledge differentiation and knowledge credibility. Specifically, the items were reworded to fit in the current research context and the team-level assessment. In addition, one item (i.e., "I know which team members have expertise in specific areas") in the specialization scale was dropped because its wording is more consistent with the definition of knowledge location than that of knowledge differentiation. To take up its position, a new item (shown as

KD5 in Table 1) was created based on the definition of knowledge differentiation. All items use a 7-point Likert scale anchored from "strongly disagree" to "strongly agree." Table 1 shows the items and their sources.

Table 1. Knowledge Differentiation and Credibility Items Used in this Study

Construct	Item Wording and Code	Source
Knowledge differentiation (KD)	• Each team member has specialized knowledge of some aspect of our project (KD1) • Different team members are responsible for different domains of expertise needed for the project (KD2) • Each team member has knowledge about some aspect of the project that no other team member on the team has (KD3) • The specialized knowledge of different team members is needed to complete the project tasks (KD4) • Members of the team specialize in different aspects of the project (KD5)	• Adapted from Lewis (2003) • Adapted from Lewis (2003) • Adapted from Lewis (2003) • Adapted from Lewis (2003) • Newly created based on the definition
Knowledge credibility (KCR)	• The members in our team are comfortable accepting project-relevant suggestions from other team members (KCR1) • The members in our team trust that other members' knowledge about the project is credible (KCR2) • The members in our team are confident applying the knowledge provided by other members to the project tasks at hand (KCR3) • The members in our team did not have much faith in other members' "expertise" (KCR4) • The members in our team like to double-check the knowledge provided by other members before applying it to the project tasks at hand (KCR5)	• Adapted from Lewis (2003) • Adapted from Lewis (2003) • Adapted from Lewis (2003) • Adapted from Lewis (2003) • Adapted from Lewis (2003)

As for knowledge location, Faraj and Sproull (2000) developed and validated a four-item scale to measure expertise location in the context of software development teams. Their definition of expertise location is consistent with the definition of knowledge location in this research. Therefore, this research adapted these items for measuring knowledge location. These items were reworded to fit into the current research context. In addition, because the knowledge differentiation and credibility scales have five items each, a fifth item (shown as KL5 in Table 2) was created for the knowledge location scale. All items use a 7-point Likert scale anchored from "strongly disagree" to "strongly agree." Table 2 shows the knowledge location scale for this research.

Table 2. Knowledge Location Items Used in this Study

Construct	Item Wording and Code	Source
Knowledge location (KL)	• The team has a good "map" of each other's talents and skills (KL1) • Each team member is doing the project tasks compatible with his or her task-relevant knowledge and skills (KL2) • Team members know what task-related skills and knowledge they each possess (KL3) • Team members know who on the team has specialized skills and knowledge that is relevant to their work (KL4) • If one member has a question about some aspect of the project, he or she knew who on the team to ask for the answer (KL5)	• Adapted from Faraj and Sproull (2000) • Adapted from Faraj and Sproull (2000) • Adapted from Faraj and Sproull (2000) • Adapted from Faraj and Sproull (2000) • Newly created based on the definition

Knowledge Coordination

Knowledge coordination is defined as the extent to which the members of an OSS team integrate their different domains of expertise to bear on software development tasks. Using this definition as the search guidance, a four-item scale developed by Faraj and Sproull (2000) was identified. The scale was designed to measure the extent to which software development teams are able to bring expertise possessed by their team members to bear on development tasks, and was validated through a field study. Therefore, this study adapted these items for measuring knowledge coordination. All items use a 7-point Likert scale anchored from "strongly disagree" to "strongly agree." Table 3 shows the knowledge coordination items used in this study.

Table 3. Knowledge Coordination Items Used in this Study

Construct	Item Wording and Code	Source
Knowledge coordination (KCO)	• People in our team share their special knowledge and expertise with one another (KCO1) • If someone in our team has some special knowledge about how to perform the project task, he or she is not likely to tell the other member about it (KCO2) • There is virtually no exchange of information, knowledge, or sharing of skills among members (KCO3) • More knowledgeable team members freely provide other members with hard-to-find knowledge or specialized skills (KCO4)	• Adapted from Faraj and Sproull (2000) • Adapted from Faraj and Sproull (2000) • Adapted from Faraj and Sproull (2000) • Adapted from Faraj and Sproull (2000)

Communication Quality

Communication quality is defined as the extent to which communication is timely and relevant to OSS development tasks. This definition is adopted from Stewart and Gosain (2006). They proposed a four-item scale to measure communication quality among the members of an OSS team and subsequently tested it in the field. Their results showed that one item (shown CQ5 in Table 4) was unreliable. However, for the purpose of this study, the question remained in the scale at the initial stage of survey development because unreliability may have been due to the negative wording in the original scale. Consequently, two more negatively worded items were created (shown as CQ4 and CQ6 in Table 4) for the scale. Table 4 shows all the items, which use a 7-point Likert scale anchored from "never" to "all the time."

Table 4. Communication Quality Items Used in this Study

Construct	Item Wording and Code	Source
Communication quality (CQ)	• People on this team answer each other's questions in a timely manner (CQ1) • Team members' responses to each other's questions are correct and useful (CQ2) • People on this team answer each other's questions in a thoughtful manner (CQ3) • Team members' responses to each other's questions are irrelevant (CQ4) • People on this team answer each other's questions after a long delay (CQ5) • Team members' responses to each other's questions are incorrect and useless.(CQ6)	• Adapted from Stewart and Gosain (2006) • Adapted from Stewart and Gosain (2006) • Adapted from Stewart and Gosain (2006) • Newly created based on the definition • Adapted from Stewart and Gosain (2006) • Newly created based on the definition

Table 5 summarizes all the survey items.

Table 5. The Initial Version of the Survey Items

Construct	Item Wording and Code	Source
Knowledge differentiation (KD)	• Each team member has specialized knowledge of some aspect of our project (KD1) • Different team members are responsible for different domains of expertise needed for the project (KD2) • Each team member has knowledge about some aspect of the project that no other team member on the team has (KD3)	• Adapted from Lewis (2003) • Adapted from Lewis (2003) • Adapted from Lewis (2003)

Table 5_continued

Knowledge differentiation (KD)	• The specialized knowledge of different team members is needed to complete the project tasks (KD4) • Members of the team specialize in different aspects of the project (KD5)	• Adapted from Lewis (2003) • Newly created based on the definition
Knowledge location (KL)	• The team has a good "map" of each other's talents and skills (KL1) • Each team member is doing the project tasks compatible with his or her task-relevant knowledge and skills (KL2) • Team members know what task-related skills and knowledge they each possess (KL3) • Team members know who on the team has specialized skills and knowledge that is relevant to their work (KL4) • If one member has a question about some aspect of the project, he or she knew who on the team to ask for the answer (KL5)	• Adapted from Faraj and Sproull (2000) • Adapted from Faraj and Sproull (2000) • Adapted from Faraj and Sproull (2000) • Adapted from Faraj and Sproull (2000) • Newly created based on the definition
Knowledge credibility (KCR)	• The members in our team are comfortable accepting project-relevant suggestions from other team members (KCR1) • The members in our team trust that other members' knowledge about the project is credible (KCR2) • The members in our team are confident applying the knowledge provided by other members to the project tasks at hand (KCR3) • The members in our team did not have much faith in other members' "expertise" (KCR4) • The members in our team like to double-check the knowledge provided by other members before applying it to the project tasks at hand (KCR5)	• Adapted from Lewis (2003) • Adapted from Lewis (2003) • Adapted from Lewis (2003) • Adapted from Lewis (2003) • Adapted from Lewis (2003)
Knowledge coordination (KCO)	• People in our team share their special knowledge and expertise with one another (KCO1)	• Adapted from Faraj and Sproull (2000)

Table 5_continued

Knowledge coordination (KCO)	• If someone in our team has some special knowledge about how to perform the project task, he or she is not likely to tell the other member about it (KCO2) • There is virtually no exchange of information, knowledge, or sharing of skills among members (KCO3) • More knowledgeable team members freely provide other members with hard-to-find knowledge or specialized skills (KCO4)	• Adapted from Faraj and Sproull (2000) • Adapted from Faraj and Sproull (2000) • Adapted from Faraj and Sproull (2000)
Communication quality (CQ)	• People on this team answer each other's questions in a timely manner (CQ1) • Team members' responses to each other's questions are correct and useful (CQ2) • People on this team answer each other's questions in a thoughtful manner (CQ3) • Team members' responses to each other's questions are irrelevant (CQ4) • People on this team answer each other's questions after a long delay (CQ5) • Team members' responses to each other's questions are incorrect and useless.(CQ6)	• Adapted from Stewart and Gosain (2006) • Adapted from Stewart and Gosain (2006) • Adapted from Stewart and Gosain (2006) • Newly created based on the definition • Adapted from Stewart and Gosain (2006) • Newly created based on the definition

Project Event Data

Several constructs of interest, including communication volume, team size, shared task experience and technical achievement, are operationalized by project event data. They are discussed below.

Communication Volume

Communication between OSS developers occurs frequently in the developer-mailing lists and/or commit logs (Gutwin et al., 2004; Krogh et al., 2003). Therefore, the total number of developer-mailing postings and commit logs serve as a proxy of communication volume. However, it is expected that sampled OSS teams may initiate their projects at different times. Teams with a longer history are certainly at the time advantage in terms of the number of postings and commit logs. In order to control this potential confounding effect, the number of postings and commit logs will be normalized by the length of project history.

Team Size

Team size is operationalized as the number of registered developers on a given project.

Technical Achievement

Technical achievement is defined as the extent to which an OSS team has completed identified project requests (i.e., bugs, patches, support, and new feature requests). This definition is adopted from Stewart and Gosain (2006). They argued that the completion of project requests is a proper indicator for outcome success in the OSS setting since meeting budget and system requirements is not relevant. Subsequently, they operationalized this construct as [(total project requests – project requests open)/total project requests] x 100, or zero for projects with no project requests. This research adopts this operationalization for technical achievement.

Data Collection

The data collection of this research consists of two phases. The first phase was a survey pilot, where the initial survey items discussed above were subject to conceptual validation and field testing. This phase has been completed. The results are discussed below. The second phase will deploy the survey validated during Phase I and collect relevant project event data.

Phase I – Survey Pilot

The main objective of Phase I was to validate the survey items introduced in the previous section. Although most of the survey items were largely adapted from prior studies, their reliabilities and validities have to be reestablished in the current research context. Therefore, the survey items were subjected to conceptual validation and field testing.

Conceptual Validation

The initial survey items (including 25 items shown in Table 5) were mixed up and exposed to a conceptual validation exercise (Kankanhalli, Tan, & Wei, 2005). Four IS PhD students participated in the exercise as sorters. Given the definitions of the five constructs (i.e., KD, KL, KCR, KCO, and CQ) that the items were designed to measure, each sorter was instructed to place each of the 25 items into a construct category.

The sorting process produces two measurements. First, Cohen's Kappa is calculated for each pair of sorters to assess the reliability of the sorting scheme (Moore & Benbasat, 1991). Moore and Benbasat (1991) recommend a Kappa score greater than 0.65. The inter-sorter Kappa scores achieved in this sorting exercise are all above 0.65, with an average of 0.74 (see Table 6). Thus, the items evaluated have good potential for internal consistency (Moore & Benbasat, 1991).

Table 6. Inter-Sorter Reliability

Sorter Pair	Pairwise Cohen's Kappa
Sorter#1-Sorter#2	0.70
Sorter#1-Sorter#3	0.75
Sorter#1-Sorter#4	0.75
Sorter#2-Sorter#3	0.74
Sorter#2-Sorter#4	0.69
Sorter#3-Sorter#4	0.79
Average	0.74

The second measurement computed is the percentage of items placed in the intended construct categories, commonly referred to as the hit rate. Although no cut-off score has been established for this measurement, generally speaking, a high percentage score is considered as an indicator for a reliable scale with good face validity (Moore & Benbasat, 1991). Overall, the four sorters correctly placed more than 78 percent of the items into the intended construct categories (see Table 7).

Table 7. Hit Rates

	Actual Category						
Target Category	KD	KL	KCR	KCO	CQ	**Total Items**	**Hit Rate (%)**
KD	17	1		2		20	0.85
KL		16		4		20	0.80
KCR			15	2	3	20	0.75
KCO	1			9	6	16	0.56
CQ			1		23	24	0.96
Average							0.78

However, the hit rates for the KCR and KCO categories were relatively low, particularly the KCO category. Examining the sorting result revealed that four items (KL2, KCR1, KCO3, and KCO4) were specifically problematic. Therefore, these four items were reworded. In addition, CQ6 was dropped because of its close similarity to CQ2. Four negatively-worded items (KD6, KD7, KL6, and KL7) were added according to the suggestion of one sorter. In addition, one new item (KCO5) was added to provide sufficient redundancy for the KCO category (DeVellis, 2003). As shown in Table 8, the modified survey consists of 29 items.

Table 8. The Modified Version of the Survey Items

Construct	**Item Wording and Code**

Table 8_continued

Knowledge differentiation (KD)	• Each team member has specialized knowledge of some aspect of our project (KD1) • Different team members are responsible for different domains of expertise needed for our project (KD2) • Each team member has knowledge about some aspect of our project that no other team member on the team has (KD3) • The specialized knowledge of several different members is needed to complete our project (KD4) • Members of our team specialize in different aspects of the project (KD5) • Members on our team have project-relevant knowledge that overlaps each other (KD6) • Members on our team are "generalists" (KD7)
Knowledge location (KL)	• Our team has a good "map" of each member's talents and skills (KL1) • Members on our team either volunteer for or are assigned to tasks commensurate with their task-relevant knowledge and skills (KL2) • Members on our team know what task-related skills and knowledge they each possess (KL3) • Members on our team know who has specialized skills and knowledge that is relevant to their work (KL4) • If one member has a question about some aspect of our project, this member knows who on the team she or he should ask for the answer (KL5) • Our members have a hard time identifying the experts on the team (KL6) • Our members have no idea what special knowledge and expertise other members on the team possess (KL7)
Knowledge Credibility (KCR)	• The members on our team do not have doubts on project-relevant suggestions from other members (KCR1) • The members on our team trust that the other members' knowledge about the project is credible (KCR2) • The members on our team are confident when applying the knowledge provided by other members to the project tasks at hand (KCR3) • The members on our team did not have much faith in the other members' "expertise" (KCR4) • The members on our team like to double-check the knowledge provided by other members before applying it to the project tasks at hand (KCR5)

Table 8_continued

Knowledge coordination (KCO)	• Members in our team share their special knowledge and expertise with one another (KCO1) • If someone in our team has some special knowledge about how to perform the project task, he or she is not likely to tell the other member about it (KCO2) • Members in our team virtually do not share their information, knowledge, or skills with one another (KCO3) • More knowledgeable members in our team willingly make their knowledge and expertise available to other members (KCO4) • Project tasks are completed by integrating the specialized knowledge of different members in our team (KCO5)
Communication quality (CQ)	• Members on our team answer each other's questions in a timely manner (CQ1) • Our team members' responses to each other's questions are correct and useful (CQ2) • Members on our team answer each other's questions in a thoughtful manner (CQ3) • Our team members' responses to each other's questions are irrelevant (CQ4) • Members on our team answer each other's questions after a long delay (CQ5)

Two psychology professors, both with extensive expertise in survey design, reviewed the modified items. Based on the definitions of the constructs given, they agreed that the items were ready for a field testing.

Field Testing

A Web-based survey was constructed based on the modified survey items. Appendix A contains the survey. The survey items were administered to a sample of OSS projects from Sourceforge.net. These projects were selected from two categories: Clustering and SysAdmin, based on the three selection criteria discussed in the "Research Site and Sample" section. One-hundred-forty-nine (149) projects were identified, and the administrators of these projects were invited to pilot-test the items. OSS project administrators were the selected sample because they are “more familiar with the team’s internal dynamics, activities, and accomplishments” than other OSS team members (Stewart & Gosain, 2006, p. 299) and are in the best position to assess team-level perceptions such as TMS, knowledge coordination, and communication quality.

However, many OSS teams have more than one administrator. Because of the practical difficulty of accessing all administrators of such an OSS team, the administrator with the longest history with the team was considered as the key informant. If that administrator did not respond, the administrator having the second longest history was considered. This selection process continued down the line until an administrator agreed to answer the survey.

The total of 283 administrators from the 149 sampled projects was invited to participate in the study. Of those invited, 72 project administrators from 61 projects completed the survey, yielding an individual-level response rate of 25.44% and a team-level response rate of 40.94%. The survey was administered through a series of three emails (see Appendix C) using Surveymonkey.com, over a period of two weeks in October 2007:

1. A brief prenotice email was sent to the administrators a few days prior to the survey administration day. It notified the administrators that a survey for a research project would arrive in a few days and that the administrator’s responses would be appreciated.
2. The second email was sent out a few days after the prenotice email. This email included a web survey link and an access code. Each administrator was provided with a unique access code. The purpose of the access code was to prevent other people from answering the survey and to help track those who responded to the survey and those who did not. A detailed cover letter (see Appendix C) was presented in the email, explaining what the survey was about, why a response was important, and how confidentiality was protected.

The enclosed link led the administrators to the Web-based survey that was hosted by Surveymonkey.com.

3. A few days after the second email, a reminder email was sent to the administrators who had not yet completed the survey. This email urged them to fill out the survey soon.

Using the survey data collected, the items were first subjected to reliability assessment. A scale with an alpha of 0.7 is considered adequately reliable (Nunnally, 1978). However, DeVellis (2003) advises that an alpha greater than 0.7 might be more desirable during the scale development stage because "some of the apparent covariation among items may be due to chance" (DeVellis, 2003, p. 96). Adopting DeVellis's view, this study attempted to bring each scale's alpha up closer to 0.8. To achieve this goal, items were dropped from the following scales due to their poor item-scale correlations: knowledge differentiation (KD6 and KD7), knowledge location (KL2), knowledge credibility (KCR1 and KCR5), knowledge coordination (KCO2 and KCO5), and communication quality (CQ4 and CQ5). Table 9 shows the reliabilities of all the scales and the number of items per scale.

Table 9. Reliability of Constructs

Construct	Number of Items	Cronbach's Alpha
KD	5	0.80
KL	6	0.88
KCR	3	0.79
KCO	3	0.82
CQ	3	0.79

The remaining items were then subjected to an exploratory factor analysis to assess their convergent and discriminant validities. A ratio of 5 responses per item is recommended for a stable factor analysis (Stevens, 1996; Tinsley & Tinsley, 1987). However, more recent studies (e.g., Kirsch, Sambamurthy, Ko, & Purvis, 2002; Stewart & Gosain, 2006) provide and demonstrate a viable solution when the overall response-to-item ratio is lower than 5:1. The solution is dividing the items into multiple subsets so that each subset reaches the response-to-item ratio of 5:1, and then performing factor analysis on each subset. The subset should be comprised of the items measuring closely related latent variables (LVs) (Kirsch et al., 2002; Stewart & Gosain, 2006).

Using the above solution, the items of the current study are divided into two subsets. One subset includes the items measuring the three dimensions of TMS, and another includes the items measuring knowledge coordination and communication quality. The principal component analysis with the varimax rotation was then conducted on each subset. Convergent validity is established if items measuring the same LV load highly (0.55 or more) on the same factor (Kankanhalli et al., 2005). Discriminant validity is demonstrated if the cross-factor loadings have a minimum gap of 0.1 (Nunnally, 1978).

Factor analysis on items related to knowledge differentiation, knowledge location, and knowledge credibility yields three factors (see Table 10). These three factors are consistent with the three LVs that these items are designed to tap. All items from the same scale have very good loadings on the same factor, and all their cross-factor loadings have a gap greater than 0.1. Therefore, the convergent and discriminant validities of these items are established.

Table 10. Validity of TMS Items

	Factor		
Items	1	2	3
KD1	0.223	**0.707**	-0.122
KD2	0.088	**0.812**	0.101
KD3	-0.241	**0.768**	0.035
KD4	-0.241	**0.763**	-0.171

Table 10_continued

KD5	0.273	**0.697**	-0.102
KL1	**0.730**	0.248	0.211
KL3	**0.712**	0.014	0.273
KL4	**0.608**	0.114	0.449
KL5	**0.817**	0.044	-0.007
KL6	**0.833**	-0.123	0.188
KL7	**0.843**	-0.055	0.168
KCR2	0.246	-0.114	**0.757**
KCR3	0.089	-0.082	**0.915**
KCR4	0.480	-0.035	**0.634**

Factor analysis on items for communication quality and knowledge coordination yields two factors (see Table 11). These two factors concur with the two LVs that these items are designed to measure. All items have very good loadings on their intended factors with minimum cross-loadings. Therefore, the convergent and discriminant validities of the items are demonstrated.

Table 11. Validity of CQ and KCO Items

	Factor	
Items	1	2
CQ1	0.272	**0.767**
CQ2	0.177	**0.853**
CQ3	-0.008	**0.847**
KCO1	**0.887**	0.276
KCO3	**0.874**	-0.069
KCO4	**0.753**	0.264

In summary, several items (i.e., KD6 and KD7, KL2, KCR1 and KCR5, KCO2 and KCO5, and CQ4 and CQ5) were dropped from their corresponding scales because of their poor item-scale correlations. The remaining items have demonstrated good validity. Table 12 shows the final version of the survey items that will be deployed in Phase II.

Table 12. The Final Version of the Survey Items

Construct	Item Wording and Code
Knowledge differentiation (KD)	• Each team member has specialized knowledge of some aspect of our project (KD1) • Different team members are responsible for different domains of expertise needed for our project (KD2) • Each team member has knowledge about some aspect of our project that no other team member on the team has (KD3) • The specialized knowledge of several different members is needed to complete our project (KD4) • Members of our team specialize in different aspects of the project (KD5)
Knowledge location (KL)	• Our team has a good "map" of each member's talents and skills (KL1) • Members on our team know what task-related skills and knowledge they each possess (KL3) • Members on our team know who has specialized skills and knowledge that is relevant to their work (KL4) • If one member has a question about some aspect of our project, this member knows who on the team she or he should ask for the answer (KL5) • Our members have a hard time identifying the experts on the team (KL6) • Our members have no idea what special knowledge and expertise other members on the team possess (KL7)
Knowledge Credibility (KCR)	• The members on our team trust that the other members' knowledge about the project is credible (KCR2) • The members on our team are confident when applying the knowledge provided by other members to the project tasks at hand (KCR3) • The members on our team did not have much faith in the other members' "expertise" (KCR4)
Knowledge Coordination (KCO)	• Members in our team share their special knowledge and expertise with one another (KCO1) • Members in our team virtually do not share their information, knowledge, or skills with one another (KCO3) • More knowledgeable members in our team willingly make their knowledge and expertise available to other members (KCO4)

Table 12_continued

Communication quality (CQ)	• Members on our team answer each other's questions in a timely manner (CQ1) • Our team members' responses to each other's questions are correct and useful (CQ2) • Members on our team answer each other's questions in a thoughtful manner (CQ3)

Phase II – Data Collection

The main objective in Phase II is to collect the data used to test the hypotheses proposed in Chapter Three. The survey data collection effort will take place with another sample of OSS projects. This sample will be constructed from two randomly selected project categories from the fourteen categories (excluding Clustering and SysAdmin) on Sourceforge.net. The survey will be administered to project administers using the three-email series laid out above. After the survey data is collected, the project event data of responding projects will be collected from their websites.

Statistical Method

This section discusses the statistical method that will be employed to analyze the data collected in Phase II. The objective is to describe the rationales of choosing these analyses.

To test the hypotheses proposed in Chapter Three, the partial least squares (PLS) and traditional structural equation modeling (SEM) will be used. PLS has three major advantages when coming to data analysis (Chin, 1998). First, it does not assume normality and independence of the observations. Second, the measurement scale used does not need to be interval or ratio. Finally, it demands a relatively small sample size to estimate the model parameters. However, PLS does not evaluate the overall fit of a model due to its variance-based nature (Chin, 1998). Traditional SEM, on the other hand, generates a number of indexes for the model fit evaluation, using the maximum likelihood (ML) method. Therefore, SEM will be used as a complement to PLS. SmartPLS 2.0 will be used for the PLS analysis (Ringle, Wende, Will, & Hamburg, 2005, http://www.smartpls.de), and Amos 5 will be used for the SEM analysis.

The sample size required for a PLS analysis, according to the widely accepted rule of thumb (Chin, 1998), is 10 times, either (a) the number of formative indicators in the largest indicator block, or (b) the number of exogenous constructs in the most complex portion of the model being tested, whichever is greater. The research model (shown in Figure 6) in this study has no formative indicators, and its most complex portion has four exogenous constructs (i.e., KL, KD, KCR, and CQ). Therefore, the PLS analysis will require at least 40 observations. SEM, on the other hand, demands a much larger sample than PLS. Several SEM scholars state that 100 subjects are the minimum satisfactory sample size (e.g., Anderson & Gerbing, 1988; Ding, Velicer, & Harlow, 1995).

Steps in Statistical Analysis

Specifically, analysis of the data from Phase II consists of four steps described below:

1. Preparing the dataset: Some OSS projects have multiple administrators and thus more than one administrator will respond to the survey. For these projects, only the response of the most experienced administrator (among the responding administrators) will be retained in the dataset.
2. Evaluating the PLS measurement model: Even though the survey items have been validated through the pilot using factor analysis, they need to be evaluated again in the PLS methodology (Stewart & Gosain, 2006). Reliability in PLS can be assessed by the average variance extracted (AVE) (Fornell & Larcker, 1981). It measures the average amount of variance shared between a latent construct (e.g., KD, KL, and KCR) and its

survey items (Stewart & Gosain, 2006). The AVE of 0.50 or greater indicates the sufficient reliability of the survey items (Chin, 1998). Furthermore, the AVE can also be used to assess discriminant validity in PLS. It is recommended that the correlations between latent constructs should be less than the square root of their AVE (Agarwal & Karahanna, 2000).

3. Evaluating the overall fit of the research model: Before testing the individual hypotheses, the overall fit of the research model has to be evaluated first. Specifically, Nevitt and Hancock (2001) recommended the Bollen-Stine adjusted p-value, with a cutoff value of 0.5. In addition, Hu and Bentler (1998, 1999) maintained that CFI, IFI, TLI, with a cutoff value of 0.95, and RMSEA, with a cutoff value of 0.6, are fairly reliable model fit indexes. Therefore, the Bollen-Stine p-value, CFI, IFI, TLI, and RMSEA, will be used in this study to evaluate the overall model fit.
4. Testing the hypotheses: PLS and SEM will be used to test the proposed hypotheses by evaluating their corresponding paths shown in Figure 6. Specifically, PLS will evaluate the paths through a resampling technique, Bootstrapping. This will generate estimates of path coefficients and their standard errors. P-values can thus be derived. On the other hand, SEM will estimate path coefficients and their standard errors, using the ML method. Corresponding P-values can then be computed. Since all the hypotheses are directional, this research will adopt the significant level of 0.05, one-tail (Freedman, Pisani, & Purves, 1998). Table 13 outlines the hypotheses to be tested, the data analysis method, and data used for each hypothesis.

Table 13. The Hypothesis Testing

Hypothesis	Data Analysis Method	Data

Table 13_continued

H1 a: Communication volume is positively associated with knowledge differentiation in an OSS team.	PLS (Bootstrapping) SEM (ML)	Dependent Variable: knowledge differentiation • KD1 • KD2 • KD3 • KD4 • KD5 Independent Variable: communication volume • The total number of developer-mailing-list postings and commit logs normalized by the length of project history
H1 b: Communication volume is positively associated with knowledge location in an OSS team.	PLS (Bootstrapping) SEM (ML)	Dependent Variable: knowledge location • KL1 • KL3 • KL4 • KL5 • KL6 • KL7 Independent Variable: communication volume • The total number of developer-mailing-list postings and commit logs normalized by the length of project history
H1 c: Communication volume is positively associated with knowledge credibility in an OSS team.	PLS (Bootstrapping) SEM (ML)	Dependent Variable: knowledge credibility • KCR2 • KCR3 • KCR4 Independent Variable: communication volume • The total number of developer-mailing-list postings and commit logs normalized by the length of project history

Table 13_continued

H2 a: The size of an OSS team is negatively associated with knowledge differentiation in an OSS team.	PLS (Bootstrapping) SEM (ML)	Dependent Variable: knowledge differentiation • KD1 • KD2 • KD3 • KD4 • KD5 Independent Variable: the size of an OSS team • The number of registered developers on a given project.
H2 b: The size of an OSS team is negatively associated with knowledge location in an OSS team.	PLS (Bootstrapping) SEM (ML)	Dependent Variable: knowledge location • KL1 • KL3 • KL4 • KL5 • KL6 • KL7 Independent Variable: the size of an OSS team • The number of registered developers on a given project.
H2 c: The size of an OSS team is negatively associated with knowledge credibility in an OSS team.	PLS (Bootstrapping) SEM (ML)	Dependent Variable: knowledge credibility • KCR2 • KCR3 • KCR4 Independent Variable: the size of an OSS team • The number of registered developers on a given project.
H3 a: Knowledge differentiation is positively associated with knowledge coordination among the members of an OSS team.	PLS (Bootstrapping) SEM (ML)	Dependent Variable: knowledge coordination • KCO1 • KCO3 • KCO4

Table 13_continued

		Independent Variable: knowledge differentiation • KD1 • KD2 • KD3 • KD4 • KD5
H3 b: Knowledge location is positively associated with knowledge coordination among the members of an OSS team. .	PLS (Bootstrapping) SEM (ML)	Dependent Variable: knowledge coordination • KCO1 • KCO3 • KCO4 Independent Variable: knowledge location • KL1 • KL3 • KL4 • KL5 • KL6 • KL7
H3 c: Knowledge credibility is positively associated with knowledge coordination among the members of an OSS team.	PLS (Bootstrapping) SEM (ML)	Dependent Variable: knowledge coordination • KCO1 • KCO3 • KCO4 Independent Variable: knowledge credibility • KCR2 • KCR3 • KCR4
H4 a: Knowledge differentiation is positively associated with communication quality among members of an OSS team.	PLS (Bootstrapping) SEM (ML)	Dependent Variable: communication quality coordination • CQ1 • CQ2 • CQ3 Independent Variable: knowledge differentiation • KD1 • KD2

Table 13_continued

		• KD3 • KD4 • KD5
H4 b: Knowledge location is positively associated with communication quality among members of an OSS team.	PLS (Bootstrapping) SEM (ML)	Dependent Variable: communication quality coordination • CQ1 • CQ2 • CQ3 Independent Variable: knowledge location • KL1 • KL3 • KL4 • KL5 • KL6 • KL7
H4 c: Knowledge credibility is positively associated with communication quality among members of an OSS team.	PLS (Bootstrapping) SEM (ML)	Dependent Variable: communication quality coordination • CQ1 • CQ2 • CQ3 Independent Variable: knowledge credibility • KCR2 • KCR3 • KCR4
H5: The quality of communication among the members of an OSS team has positive effects on knowledge coordination in the team.	PLS (Bootstrapping) SEM (ML)	Dependent Variable: knowledge coordination • KCO1 • KCO3 • KCO4 Independent Variable: communication quality coordination • CQ1 • CQ2 • CQ3

Table 13_continued

H6: Knowledge coordination within an OSS team positively affects the team's technical achievement.	PLS (Bootstrapping) SEM (ML)	Dependent Variable: technical achievement • (total project requests – project requests open)/total project requests] x 100, or zero for projects with no project requests Independent Variable: knowledge coordination • KCO1 • KCO3 • KCO4
H7: The quality of communication among the members of an OSS team positively affects the team's technical achievement.	PLS (Bootstrapping) SEM (ML)	Dependent Variable: technical achievement • (total project requests – project requests open)/total project requests] x 100, or zero for projects with no project requests Independent Variable: communication quality coordination • CQ1 • CQ2 • CQ3

Preliminary Results

A preliminary hypothesis testing was conducted by using the survey data collected in Phase I. The results are discussed below.

Preparing the Dataset

In Phase I, 72 project administrators from 61 projects completed the survey. For the projects where multiple administrators answered the survey, only the response of the most experienced administrator (among the responding administrators) was retained in the dataset. As a result, the survey data has 61 observations. Next, the relevant project event data was collected from the responding projects' websites. However, quite a few projects in the sample did not have developer-mailing lists. Therefore, H1a, H1b, and H1c were only partially evaluated with the number of commit logs.

Evaluating the PLS Measurement Model

The AVE of one construct (i.e., knowledge differentiation) is less than 0.5, the recommended lower bound (Chin, 1998). Therefore, the loadings of its indicators were examined. Among the five indicators (i.e., KD1, KD2, KD3, KD4, and KD5), KD5 had the lowest loading (i.e., 0.38), and thus it was dropped to improve the corresponding AVE value.

Table 14 shows the correlations between constructs after removing KD5. Values in the diagonal are their AVE and values in parentheses are the square root of AVE. All constructs have a value of AVE greater than 0.5 and their correlations are less than the square root of their AVE. Therefore, the reliability and validity of the measurement model is established.

Table 14. The Correlations between Constructs and their AVE

Construct	CQ	KCO	KCR	KD	KL
CQ	0.68(0.82)				
KCO	0.33	0.74(0.86)			
KCR	0.48	0.33	0.69(0.83)		
KD	-0.44	-0.34	-0.32	0.53(0.73)	
KL	0.48	0.43	0.52	-0.32	0.59(0.77)

Testing the Hypotheses

The hypotheses were evaluated using the bootstrapping method with N = 61 cases (Efron, 2000) and 200 resamples (Tenenhaus, Vinzi, Chatelin, & Lauro, 2004). The results are shown in Figure 9 and Table 15 and 16. Figure 7 shows the path coefficients and their significance levels. To present a clear picture, only are the significant paths shown. Table 15 displays all the path coefficients, their t-value, and p-value. Table 16 summarizes the results of the hypotheses testing.

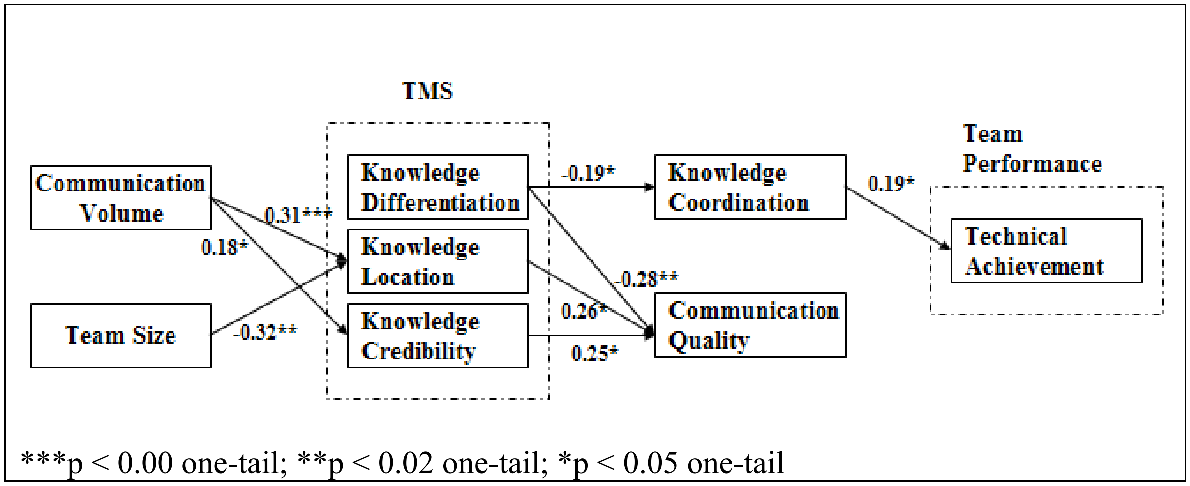


Figure 7. Significant Paths of the Research Model (Phase I)

Table 15. Path Coefficients

	Path Coefficients	**t-Value**	**p-Value (one-tail)**
Knowledge Location			
Communication Volume (commit logs only)	**0.31***	**4.39**	**0.00**
Team Size	**-0.32****	**2.24**	**0.01**
Knowledge Differentiation			
Communication Volume (commit logs only)	0.01	0.14	0.45
Team Size	0.07	0.56	0.29
Knowledge Credibility			
Communication Volume (commit logs only)	**0.18***	**1.79**	**0.04**
Team Size	-0.02	0.17	0.43
Communication Quality			
Knowledge Location	**0.26***	**1.87**	**0.03**
Knowledge Differentiation	**-0.28****	**2.52**	**0.01**
Knowledge Credibility	**0.25***	**1.95**	**0.03**
Knowledge Coordination			
Knowledge Location	0.29	1.54	0.06
Knowledge Differentiation	**-0.19***	**1.73**	**0.04**
Knowledge Credibility	0.09	0.50	0.31
Communication Quality	0.07	0.30	0.38

Table 15_continued

Technical Achievement			
Knowledge Coordination	**0.19***	**1.65**	**0.05**
Communication Quality	0.02	0.12	0.45

Table 16. Summary of Hypotheses Testing (Phase I)

Hypothesis	Results
H1a. Communication Volume (commit logs only) → Knowledge Differentiation	Not Supported
H1b. Communication Volume (commit logs only)→ Knowledge Location	Supported
H1c. Communication Volume (commit logs only) → Knowledge Credibility	Supported
H2a. Team Size → Knowledge Differentiation	Not Supported
H2b. Team Size → Knowledge Location	Supported
H2c. Team Size → Knowledge Credibility	Not Supported
H3a. Knowledge Differentiation → Knowledge Coordination	Not Supported* (* = significant but opposite sign)
H3b. Knowledge Location → Knowledge Coordination	Not Supported
H3c. Knowledge Credibility → Knowledge Coordination	Not Supported
H4a. Knowledge Differentiation → Communication Quality	Not Supported* (* = significant but opposite sign)
H4b. Knowledge Location → Communication Quality	Supported
H4c. Knowledge Credibility → Communication Quality	Supported
H5. Communication Quality → Knowledge Coordination	Not Supported
H6. Knowledge Coordination → Technical Achievement	Supported
H7. Communication Quality → Technical Achievement	Not Supported

Lessons Learned from the Pilot Study

Several lessons are learned from conducting the pilot study. These lessons, as discussed below, help refine the survey items and sample selection criteria.

Revising the Survey Items

The results from the pilot study showed that several items in Table 8 were not reliable (i.e., KD5, KD6, and KD7, KL2, KCR1 and KCR5, KCO2 and KCO5, and CQ4 and CQ5) and thus were dropped from their corresponding scales. The Web-based survey is modified accordingly to reflect the changes. Appendix B contains the modified survey.

Revising Sample Selection Criteria

Three criteria, as discussed in “Research Site and Sample” Section, were applied when selecting the sample of OSS projects for the pilot study. However, some projects in the sample do not have developer-mailing lists, and thus H1a, b, and c were not tested. Therefore, the fourth criterion will be added when selecting the sample for the next round of data collection (i.e., the Phase II data collection). That is, the OSS projects to be included in the sample must have developer-mailing lists.

The next chapter will describe the data collected at Phase II and the results of analysis. Chapter Six will discuss the theoretical and pragmatic implications of the findings, the limitations of the research, and suggests future research directions.

CHAPTER FIVE: ANALYSES AND RESULTS

This chapter presents the analysis results derived from data gathered in Phase II. It begins with a description of the data collection process (Phase II) and then provides descriptive statistics. Next, the reliability and validity of the measurement model are discussed. This is followed by hypothesis testing: the research model is first evaluated as a whole using the conventional model fit indexes, and individual hypotheses are then tested by evaluating the statistical significance of their corresponding paths in the research model.

Sample Data Collection (Phase II)

Data for Phase II are gathered from two project categories on Sourceforge.net: Networking and Development. Two-hundred-sixteen (216) projects are identified using the selection criteria previously discussed in Chapter Four. Ninety-three (93) projects are from the Networking category, and one-hundred-twenty-three (123) are from the Development category. The administrators of these projects are invited to participate in the study by a series of emails (see Appendix C). The first series was administered in November 2008. In February 2009, follow-up emails were sent to project administrators who had not responded to the first series. A total of 586 administrators from 216 projects are invited to participate in the study. Of those invited, one-hundred-fifty-five (155) administrators from 103 projects respond to the survey. However, eight projects are removed from the dataset due to lack of sufficient information. As a result, the responses of 147 administrators from 95 projects are valid, yielding an individual response rate of 25.09% and a team-level response rate of 43.98%. For each responding project, only the response of the most experienced administrator is retained in the final dataset. In addition to the survey data, relevant project event data, such as the number of postings in the developer-mailing lists and the number of commit logs, are collected from the responding projects' websites.

Sample Descriptive Statistics

This section reports the basic demographic information of sampled projects as well as descriptive statistics for the project event and survey data.

Demographic Information

The demographic information of the sampled projects is described in terms of project category, project history, and team size. As shown in the following sections, sampled projects are quite diversified in these aspects.

Project Category

Table 17 shows the distribution of sampled projects across the Development and Networking project categories. Fifty-nine projects (62.11%) in the sample are from the Development category, and thirty-six projects (37.89%) in the sample are from the Networking category. The larger number of Development projects in the sample is expected since the Development category has a much larger project population (i.e., 54,964 projects) than the Networking category (i.e., 7,702 projects).

Table 17. Project Category

	Frequency	Percent
Development	59	62.11
Networking	36	37.89
Total	95	100.0

Project History

As Table 18 shows, project history ranges from 14 months to 108 months, with an average value of approximately 68 months.

Table 18. Descriptive Statistics – Project History

	Minimum	Maximum	Mean	Std. Deviation
Project History (in months)	14	108	68	28

Team Size

As Table 19 shows, the number of developers in the sampled projects ranges from two (2) to 93, with an average team size is 15 developers.

Table 19. Descriptive Statistics – Team Size

	Minimum	Maximum	Mean	Std. Deviation
Team Size	2	93	15	16

Descriptive Statistics for the Project Event Data

Sourceforge.net archives a large quantity of the project event data. Such data is publicly accessible from each OSS project's website. This study specifically collects the number of postings in the developer-mailing lists, the number of commit logs, and the number of project issues, from responding projects' websites to measure several variables of interest discussed below.

Communication Volume

Communication in an OSS team takes place primarily in the developer-mailing list and through commit logs (Gutwin et al., 2004, Krogh et al., 2003). Therefore, communication volume is measured by summing the number of postings in the developer-mailing lists and commit logs. A large percentage of projects in the sample (i.e., about 45%) have equivalent numbers of postings and commit logs. About nineteen percent of the projects appear to rely more heavily on the developer-mailing lists for communication, whereas about thirty-six percent of the projects depend more on commit logs for communication. However, the reliance on developer-mailing lists or commit logs seems random. There is no pattern in usage of one over the other. Table 20 shows that communication volume ranges from 52 to 91990 messages, with an average value of 7366 messages.

Table 20. Descriptive Statistics – Communication Volume

	Minimum	Maximum	Mean	Std. Deviation
Communication Volume	52	91990	7366	12867

Technical Achievement

Technical achievement refers to the extent to which an OSS team has completed software development tasks (Stewart & Gosain, 2006), and is operationalized as the percentage of project issues resolved, such as bugs, feature requests, and patches. Table 21 shows that technical achievement ranges from 2% to 100%, with an average value of 75%. Although one project has only completed 2% of the project issues reported, it has a history close to 73 months and 467 messages exchanged among the developers. Therefore, it has enough data to be included in the sample.

Table 21. Descriptive Statistics – Technical Achievement

	Minimum	Maximum	Mean	Std. Deviation
Technical Achievement	2%	100%	75%	21%

Descriptive Statistics for the Survey Items

Several variables of interest, including communication quality (CQ), knowledge differentiation (KD), knowledge location (KL), knowledge credibility (KCR), and knowledge coordination (KCO), are measured using the survey scales verified in the pilot study.

Specifically, communication quality refers to the extent to which communication is timely and relevant to OSS development tasks (Stewart & Gosain, 2006) and is measured using three survey items: CQ1, CQ2, and CQ3. Knowledge differentiation refers to the extent to which the developers of an OSS team specialize in different knowledge domains (Lewis, 2003; Palazzolo et al., 2006; Wegner, 1987), and is measured using four survey items: KD1, KD2, KD3, and KD4. Knowledge location refers to the extent to which the developers of an OSS team are familiar with the distribution of task relevant knowledge within the team (Wegner, 1987; Wegner et al., 1985), and is measured using six survey items: KL1, KL3, KL4, KL5, KL6, and KL7. Knowledge credibility refers to the extent to which the developers of an OSS team have confidence in each other's knowledge (Lewis, 2003; Moreland, 1999), and is measured using three survey items: KCR2, KCR3, and KCR4. Knowledge coordination refers to the extent to which the members of an OSS team integrate their different domains of expertise on software development tasks (Faraj & Sproull, 2000; Mitchell, 2006; Tiwana, 2004), and is measured using three survey items: KCO1, KCO3, and KCO4.

Table 22 presents descriptive statistics for the survey items. Mean values for most items are above 4 (i.e., the midpoint on the Likert scale), indicating the possibility of positive bias in the survey data. However, this is somewhat anticipated considering the majority of sampled projects have resolved over 50% of the reported project issues.

Table 22. Descriptive Statistics – Survey Items

Survey Topic	Survey Item	Mean	Std. Deviation
Communication Quality	CQ1	5.40	1.34
	CQ2	5.69	1.13
	CQ3	5.65	1.21
Knowledge Differentiation	KD1	5.31	1.21
	KD2	4.82	1.53
	KD3	3.96	1.66
	KD4	4.51	1.84
Knowledge Location	KL1	4.71	1.35
	KL3	5.16	1.22
	KL4	5.35	1.10
	KL5	5.58	1.08
	KL6	5.62	1.25
	KL7	5.78	1.36
Knowledge Credibility	KCR2	5.94	.84
	KCR3	5.87	.85
	KCR4	6.06	.93
Knowledge Coordination	KCO1	5.91	1.01
	KCO3	6.13	1.14
	KCO4	5.83	1.11

Testing Differences across Project Categories

Data are obtained from two different project categories: Networking and Development. Hence, the ANOVA test is conducted to determine whether dependent variables are significantly different across project categories (Stewart & Gosain, 2006). The results (shown in Appendix D) indicate no significant difference across project categories. Therefore, the two sets of projects are pooled for the subsequent analysis.

Measurement Model Assessment

The psychometric properties of the measurement model need to be confirmed before testing hypotheses (Anderson & Gerbing, 1988). Therefore, this section discusses the validity and reliability of the measurement model used in the research.

Validity

The validity is assessed by the loadings and cross-loadings of the survey items: (1) the items should load highly (i.e., greater than 0.71) on their intended constructs (Chin, 1998), and (2) the gaps between the cross-loadings should be greater than 0.1 (Nunnally, 1978). Using the criteria above, two items (i.e., KD3 and KD4) in the scale for knowledge differentiation are removed due to their low loadings (i.e., 0.49 and 0.35, respectively). Table 23 shows the loadings and cross-loadings of the remaining items. All of them load highly on their intended constructs, and the cross-loadings have sufficient gaps.

Table 23. Loadings and Cross-Loadings

	CQ	KCO	KCR	KD	KL
CQ1	**0.89**	0.50	0.44	0.31	0.43
CQ2	**0.90**	0.41	0.49	0.26	0.44
CQ3	**0.95**	0.47	0.54	0.22	0.46
KCO1	0.48	**0.93**	0.54	0.26	0.40
KCO3	0.46	**0.90**	0.58	0.37	0.52
KCO4	0.38	**0.82**	0.47	0.15	0.26
KCR2	0.53	0.60	**0.93**	0.36	0.56
KCR3	0.50	0.52	**0.92**	0.35	0.49
KCR4	0.40	0.50	**0.84**	0.39	0.55
KD1	0.28	0.33	0.43	**0.91**	0.46
KD2	0.22	0.19	0.28	**0.86**	0.46
KL1	0.32	0.30	0.43	0.41	**0.78**
KL3	0.34	0.36	0.46	0.42	**0.83**
KL4	0.49	0.35	0.53	0.53	**0.87**
KL5	0.35	0.46	0.44	0.37	**0.77**
KL6	0.43	0.25	0.47	0.27	**0.79**
KL7	0.40	0.46	0.51	0.48	**0.81**

Further evidence of validity is obtained by evaluating the AVE of the survey scales. As shown in Table 24, the square root of the AVE (i.e., values in the diagonal) is greater than the correlations between latent variables, indicating the validity of the measurement model (Fornell & Larcker, 1981).

Table 24. Correlations and Square Root of AVE

	CQ	KCO	KCR	KD	KL
CQ	0.91				
KCO	0.50	0.88			
KCR	0.54	0.60	0.90		
KD	0.29	0.30	0.41	0.88	
KL	0.49	0.46	0.59	0.52	0.81

Reliability

The reliability of the measurement model is evaluated in terms of AVE, composite reliability, and Cronbach's Alpha (Chin, 1998). Table 25 shows that all the scales are above the recommended level (composite reliability and Cronbach's Alpha greater than 0.7 and AVE greater than 0.5). Therefore, the reliability of the measurement model is established.

Table 25. Reliabilities of Scales

	AVE	Composite Reliability	Cronbach's Alpha
CQ	0.84	0.94	0.90
KCO	0.78	0.91	0.86
KCR	0.81	0.93	0.88
KD	0.78	0.88	0.73
KL	0.65	0.92	0.89

Hypothesis Testing

This section discusses the methods used for hypothesis testing, and reports model fit indexes and results of hypothesis testing.

Testing Methods

Hypotheses are tested using both PLS and SEM. Although PLS has advantages in terms of data distribution and sample size, it does not produce indexes for evaluating the overall model fit (Chin, 1998). On the other hand, SEM, with the maximum likelihood (ML) estimation method, generates a number of indexes for the model fit evaluation. Therefore, SEM is used here as a complement to PLS. However, SEM demands a large sample size. Several SEM scholars (e.g., Anderson & Gerbing, 1988; Ding et al., 1995) state that 100 subjects are the minimum satisfactory sample size. Data gathered here has only 95 projects. Nevertheless, the results of hypothesis testing from PLS and SEM are virtually identical, as shown in subsequent sections. Therefore, the SEM results should be trusted even though the sample size is 5 subjects short from the recommended 100 subjects.

PLS uses the bootstrapping resampling technique to estimate standard errors and path coefficients, and the hypotheses are then assessed by examining the T statistics of corresponding paths. SEM, on the other hand, uses the ML method to produce the estimates of standard errors and path coefficients. The hypotheses are then evaluated by examining the statistical significance of their corresponding paths. If the path is significant at the 0.05 level and the sign of the path coefficient is consistent with the direction hypothesized, the corresponding hypothesis is considered supported.

Model Evaluation

Before examining the individual paths, the overall fit of the model has to be evaluated. A number of indexes have been proposed. For example, Nevitt and Hancock (2001) recommended the Bollen-Stine adjusted p-value, with a cutoff value of 0.5, rather than the traditional χ^2 to assess the model fit. χ^2 leads to an excessively high rate of rejecting a model even though it is not statistically significantly different from the true model, that is, Type II error (Chou & Bentler, 1995; West, Finch, & Curran, 1995). In addition, Hu and Bentler (1998, 1999) evaluated the performance of GFI, AGFI, RMSEA, CFI, IFI, and TLI. Based on the findings, they concluded that GFI and AGFI perform poorly and should not be used for model fit assessment; on the other hand, CFI, IFI, TLI, with a cutoff value of 0.95, and RMSEA, with a cutoff value of 0.6, are fairly reliable. Therefore, the Bollen-Stine p-value, CFI, IFI, TLI, and RMSEA, are used in this study to evaluate the overall model fit. Table 26 reports these indexes. All fit indexes are above the recommended values, and thus the model has a good fit.

Table 26. Model Fit Indexes

Model Fix Index	Value
Bollen-Stine p-value	0.32
CFI	0.96
IFI	0.96
TLI	0.95
RMSEA	0.05

Testing Hypotheses

The hypotheses are tested by evaluating the statistical significance of corresponding paths shown in the research model (i.e., Figure 6). For easy reference, figure 6 is re-presented below. The results of hypothesis testing are discussed below.

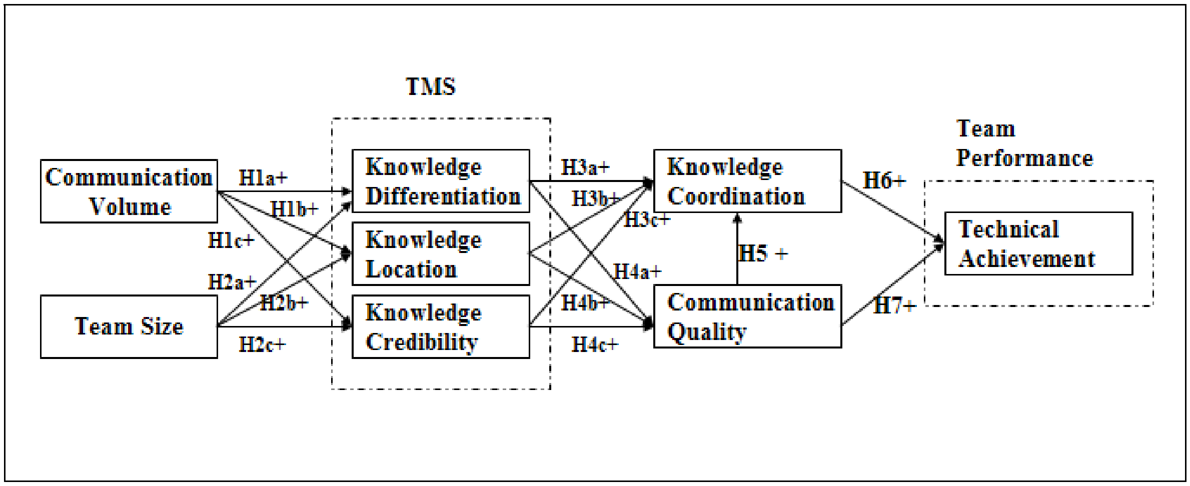


Figure 6. The Research Model

Hypothesis 1 (H1)

H1 states that communication volume is positively associated with the TMS development in an OSS team. Communication in an OSS team takes place primarily in the developer-mailing lists and through commit logs (Gutwin et al., 2004, Krogh et al., 2003). Therefore, communication volume is measured by summing the number of postings in the developer-mailing list and commit logs. TMS, on the other hand, is measured using the survey answers from project administrators.

Since TMS is defined as a three-dimensional construct (i.e., knowledge differentiation, knowledge location, and knowledge credibility), H1 encompasses three sub-hypotheses (i.e., H1a, H1b, and H1c). The testing results of these sub-hypotheses are discussed below.

H1a states that there is a positive relation between communication volume and knowledge differentiation; H1b states that there is a positive relation between communication volume and knowledge location; H1c states that there is a positive relation between communication volume and knowledge credibility. These sub-hypotheses are tested by evaluating the statistical significance of the paths corresponding to H1a, H1b, and H1c. The dependent variables are knowledge differentiation, knowledge location, and knowledge credibility, and the independent variable is communication volume. Table 27 provides the results.

Table 27. Hypothesis Testing – H1 (H1a, H1b, and H1c)

Hypothesis	Testing Method	Path Coefficient	Standard Error	T Statistics	P-value (one-tail)
H1a	PLS	0.03	0.09	0.35	0.36
	SEM	0.03	0.00	0.22	0.41
H1b	PLS	0.13	0.10	1.24	0.11
	SEM	0.12	0.00	0.99	0.16
H1c	PLS	0.12	0.09	1.42	0.08
	SEM	0.11	0.00	0.93	0.18

As shown, none of the sub-hypotheses for H1 are supported. These sub-hypotheses were proposed based on the reasoning that team members can negotiate knowledge responsibilities, exchange information about each other's domains of expertise, and assess the credibility of each other's expertise recognition, through intensive communication (Wegner, 1987, Wegner et al., 1985).

This reasoning has been generally supported in colocated teams (e.g., Hollingshead, 1998a; Lewis, 2004), but has not obtained sufficient validation in virtual teams. Only one prior study (i.e., Kanawattanachai & Yoo, 2007) looked into the relation between communication volume and TMS in virtual teams. They found that the relation is positive and statistically significant, which is not consistent with the finding reported here.

The contextual differences of the teams may explain the inconsistent findings between this study and the study of Kanawattanachai and Yoo (2007). Kanawattanachai and Yoo (2007) researched the virtual teams of MBA students performing a business simulation game. The average number of mailing-list postings was about 136 messages. On the other hand, this study examines virtual teams of OSS developers conducting software development projects. Due to the nature of software development, communication volume is much higher in OSS project teams. The descriptive statistics show that the mean value for communication volume is 7,366 messages. About thirty-two percent of the sample projects have communication volume greater than 5,000 messages.

H1 proposed hinges on the assumption that OSS developers read most or all of electronic messages, including archived ones. Given such high volume of messages, OSS developers probably do not have time to read most of them, and instead only pay attention to the most recent postings and commit logs, or ones that seem especially relevant to them. As a result, communication through postings and commit logs might not be able to furnish team members with a clear picture about TMS of the team, and thus none of sub-hypotheses in H1 are supported in the context of OSS project teams.

Hypothesis 2 (H2)

H2 states that team size is negatively associated with the TMS development in an OSS team. Team size is operationalized as the number of registered developers on a given project. TMS, on the other hand, is measured using the survey answers from project administrators.

Again, since TMS is defined as a three-dimensional construct (i.e., knowledge differentiation, knowledge location, and knowledge credibility), H2 encompasses three sub-hypotheses. H2a states that there is a negative relation between team size and knowledge differentiation. H2b states that there is a negative relation between team size and knowledge location, and H2c states that there is a negative relation between team size and knowledge credibility. Table 28 provides the results.

Table 28. Hypothesis Testing – H2 (H2a, H2b, and H2c)

Hypothesis	Testing Method	Path Coefficient	Standard Error	T Statistics	P-value (one-tail)
H2a	PLS	0.17	0.10	1.67	0.05
	SEM	0.21	0.01	1.55	0.06

Table 28_continued

H2b	PLS	0.01	0.12	0.12	0.45
	SEM	0.08	0.01	0.67	0.25
H2c	PLS	0.09	0.13	0.70	0.24
	SEM	0.11	0.01	0.88	0.19

As shown, the path coefficients for H2a from both analyses are positive, and p values for H2a are low and similar with both PLS and SEM (.05 versus .06). However, H2a proposed a negative relation between team size and the knowledge differentiation dimension of TMS, because members in a large OSS team might find it difficult identifying the expertise of other members and the uniqueness of their expertise (Austin, 2003; Palazzolo et al., 2006). Thus H2a is not supported.

The early research on TMS was focused on small groups of two to three people in laboratories (Liang et al., 1995; Moreland, 1999; Hollingshead, 1998a, 1998b; Wegner et al., 1991). Several field studies later examined TMS in teams of 3 to 11 members (Austin, 2000; Kanawattanachai & Yoo, 2007). These studies all confirmed that TMS is able to develop in such groups, but none of them directly addressed the nature of the relation between team size and TMS.

Using computational models, Palazzolo et al. (2006) ran a simulation to look specifically into the relation between team size and the knowledge differentiation dimension of TMS. They reported a negative sign for the relation, which is contradictory to the finding here. However, Palazzolo et al. (2006) cautioned that the nature of the relation between team size and knowledge differentiation might be contingent upon task complexity. As the complexity of the group task increases, more members need to be recruited to take charge of diverse domains of knowledge needed. As a result, the relation between team size and knowledge differentiation might turn to be positive. Therefore, the positive coefficient for H2a, as reported in Table 29, might be related to the high complexity of OSS projects sampled in this study.

H2b and H2c are based on the reasoning that members in a large OSS team might find it difficult identifying the expertise of other members, the uniqueness of a member's expertise, and the quality of others' expertise (Austin, 2003; Palazzolo et al., 2006). Note that p values for H2b and H2c are far greater than 0.05 when applying either PLS or SEM. Therefore, neither H2b nor H2c is supported, suggesting team size has no effects on either knowledge location or knowledge credibility.

One possible explanation for these results might be based on the peculiar structure of OSS teams. Several previous studies (e.g., Crowston & Howison, 2006; Mockus et al., 2002; Raymond, 2000) reported that an OSS project team is typically composed of two sub-groups: core and ancillary developers. Core developers are largely responsible for design decisions and major coding tasks. In order to reach consensus on design decisions and prevent code conflicts, they must interact constantly and intensively with each other (Crowston & Howison, 2006). Ancillary developers, on the other hand, take charge of supporting tasks, such as testing new releases, reporting bugs, and occasionally contributing some code. Interactions among them are commonly sparse (Raymond, 2000). Team size in this study is operationalized as the number of registered developers on a given project, thus including both core and ancillary developers. However, it might be just the number of core developers that is associated with knowledge location and knowledge credibility. Data about the number of core developers is not available because both core and ancillary developers are officially titled as "developers" in OSS project teams. Therefore, this postulation can not be empirically tested. Nevertheless, it remains as a potential explanation for why H2b and H2c are not supported.

Hypothesis 3 (H3)

A fully developed TMS provides a team with a broad knowledge base (Wegner, 1987, Wegner et al., 1985), directs knowledge seekers to where knowledge is located (Wegner et al., 1985, Wegner, 1987, Moreland, 1999), and makes a member readily accept the knowledge from others (Joshi and Sarker, 2006, Joshi et al., 2004, Joshi et al., 2005). Therefore, H3 states that TMS is positively associated with knowledge coordination among the members of an OSS team. Both knowledge coordination and TMS are measured using the survey answers from project administrators.

H3 encompasses three sub-hypotheses. H3a states that knowledge differentiation is positively associated with knowledge coordination among the members of an OSS team. H3b states that knowledge location is positively associated with knowledge coordination among the members of an OSS team, and H3c states that knowledge credibility is positively associated with knowledge coordination among the members of an OSS team. These sub-hypothesis are tested by evaluating the statistical significance of the paths corresponding to H3a, H3b, and H3c. The dependent variable is knowledge coordination, and the independent variables are knowledge differentiation, knowledge location, and knowledge credibility. Table 29 provides the results.

Table 29. Hypothesis Testing – H3 (H3a, H3b, and H3c)

Hypothesis	Testing Method	Path Coefficient	Standard Error	T Statistics	P-value (one-tail)
H3a	PLS	0.02	0.10	0.19	0.43
	SEM	0.05	0.15	0.32	0.37
H3b	PLS	0.08	0.14	0.60	0.27
	SEM	0.05	0.13	0.30	0.38
H3c	PLS	0.42	0.16	2.63	0.00
	SEM	0.47	0.16	3.44	0.00

The results from both testing methods show that the relation between knowledge differentiation and knowledge coordination is not statistically significant ($p = 0.43$ with PLS and $p = 0.37$ with SEM). Therefore, H3a is not supported.

This result might be explained by modularity of software architecture adopted by many OSS projects (Krogh et al., 2003; Osterloh & Rota, 2007). A modular project consists of several relatively independent components or modules (Narduzzo & Rossi, 2003). Interfaces between modules are well-defined, and thus coordination and integration among developers are reduced to the minimal level (Shanchez & Mahoney, 1996). Krogh et al. (2003) reported that the majority of developers tend to focus on one or two specific modules, that is, specializing in certain domains. The modularized software architecture minimizes coordination need between developers from different modules (Osterloh & Rota, 2007). This may account for why the proposed relation between knowledge differentiation and knowledge coordination was not supported.

The results from both testing methods show that the relation between knowledge location and knowledge coordination is not statistically significant ($p = 0.27$ with PLS and $p = 0.38$ with SEM). Therefore, H3b is not supported.

Knowledge location measured in this study is essentially the perception that OSS developers have about who has what knowledge in a project team. Austin (2003) argued that team members' perception about knowledge location does not necessarily reflect the actual location of knowledge. By surveying 27 groups in a large apparel and sporting goods company, he found that the perception about who possesses what knowledge has positive impacts on team performance only when the perception is accurate. Kanawattanachai and Yoo (2007) had a similar finding. They studied the relation between the perception of knowledge location and knowledge coordination in 38 virtual teams. Each team had 3 to 4 members, and each member was assigned to one of four expertise domains. Hence, it is fairly easy to identify who has what expertise in the team. In other words, perceived knowledge location should be very close to actual knowledge location in such context. As a result, they reported a positive and statistically significant coefficient for the relation.

The fact that H3b is not supported in this study may be because of inaccuracy of perceived knowledge location. After all, software development is a group task involving much complexity. Furthermore, voluntary nature of the workforce and non-face-to-face communication make it harder to identify the actual location of knowledge. Inaccuracy of perceived knowledge location will certainly not facilitate knowledge coordination.

The results from both testing methods show that the relation between knowledge credibility and knowledge coordination is positive and statistically significant ($\beta = 0.42$, $p = 0.00$ with PLS and $\beta = 0.47$, $p = 0.00$ with SEM). Therefore, H3c is supported. This finding is consistent with the attribution theory (Kelley, 1973). The theory maintains that recipients of knowledge assess credibility of knowledge source. Only when the source is credible, knowledge itself will be perceived to be useful, thus assisting knowledge coordination. Ko et al. (2005) studied knowledge coordination between ERP implementing organizations and external consultants, and reported a positive relation between source credibility and knowledge coordination. Sarker et al. (2005) also had similar finding in software teams made of university students. The results shown in Table 30 indicate that the same relation remains true in the OSS project settings.

Hypothesis 4 (H4)

H4 states that TMS is positively associated with communication quality among members of an OSS team. This hypothesis is based on the reasoning that knowledge differentiation enables detailed and most relevant knowledge to be communicated when dealing with critical development tasks (Wegner, 1987, Wegner et al., 1985). Further, knowledge location and credibility assist the developers in determining from whom they can acquire needed knowledge (Lewis, 2003; Moreland, 1999). Both communication quality and TMS are measured using the survey answers from project administrators.

Since TMS is defined as a three-dimensional construct (i.e., knowledge differentiation, knowledge location, and knowledge credibility), H4 encompasses three sub-hypotheses (i.e., H4a, H4b, and H4c). The testing results of these sub-hypotheses are discussed below.

H4a states that knowledge differentiation is positively associated with communication quality among the members of an OSS team. H4b states that knowledge location is positively associated with communication quality among the members of an OSS team, and H4c states that knowledge credibility is positively associated with communication quality among the members of an OSS team. The dependent variable is communication quality, and the independent variable is knowledge differentiation, knowledge location, and knowledge credibility. Table 30 provides the results.

Table 30. Hypothesis Testing – H4 (H4a, H4b, and H4c)

Hypothesis	Testing Method	Path Coefficient	Standard Error	T Statistics	P-value (one-tail)
H4a	PLS	-0.00	0.09	0.04	0.48
	SEM	-0.10	0.18	0.67	0.25
H4b	PLS	0.26	0.12	2.11	0.02
	SEM	0.29	0.16	1.82	0.04
H4c	PLS	0.38	0.14	2.66	0.00
	SEM	0.47	0.18	3.71	0.00

The results from both testing methods show that the relation between knowledge differentiation and communication quality is not statistically significant ($p = 0.48$ with PLS and $p = 0.25$ with SEM). Therefore, H4a is not supported.

This result might be explained by modularity of software architecture adopted by many OSS projects (Krogh et al., 2003; Osterloh & Rota, 2007). A modular project consists of several relatively independent components or modules (Narduzzo & Rossi, 2003). Interfaces between modules are well-defined, and thus coordination and integration among developers are reduced to the minimal level (Shanchez & Mahoney, 1996). Krogh et al. (2003) reported that the majority of developers tend to focus on one or two specific modules, that is, specializing in certain domains. The modularized software architecture minimizes communication need between developers from different modules (Osterloh & Rota, 2007). This may be why the relation between knowledge differentiation and communication quality is not statistically significant.

The results from both testing methods show that the relation between knowledge location and communication quality is positive and statistically significant ($\beta = 0.26$, $p = 0.02$ with PLS and $\beta = 0.29$, $p = 0.04$ with SEM). Therefore, H4b is supported, suggesting that knowing distribution of knowledge within a team helps team members communicate effectively and efficiently. This finding is consistent with Wegner's theorization of knowledge location. He argued that knowledge location points members to the individuals who have useful information (Moreland & Levine, 1992) and guides members to communicate problems with the people most likely to solve them. Lewis (2004) found that members in MBA consulting teams spent less time searching around for information needed if they were familiar with other members' specialties.

The results from both testing methods show that the relation between knowledge credibility and communication quality is statistically significant ($\beta = 0.38$, $p = 0.00$ with PLS and $\beta = 0.47$, $p = 0.00$ with SEM). Therefore, H4c is supported, suggesting that credibility of knowledge source facilitates communication process among team members. This result is consistent with the argument (Lewis, 2004; Moreland, 1999) that trust among group members enables them to engage open communication and also reduces unnecessary communication overhead (e.g., justifying why information provided is accurate).

Hypothesis 5 (H5)

H5 states that the quality of communication among the members of an OSS team is positively associated with knowledge coordination in the team. This is because, through timely and relevant communication, the various knowledge among different members of an OSS team can be integrated on the development tasks. This hypothesis is tested by evaluating the statistical significance of the path corresponding H5. The dependent variable is knowledge coordination, and the independent variable is communication quality. Both of them are measured using the survey answers from project administrators. Table 31 provides the results.

Table 31. Hypothesis Testing – H5

Hypothesis	Testing Method	Path Coefficient	Standard Error	T Statistics	P-value (one-tail)
H5	PLS	0.23	0.12	1.90	0.03
	SEM	0.22	0.10	1.87	0.03

The results from both testing methods show that the relation between communication quality and knowledge coordination is positive and statistically significant ($\beta = 0.23$, $p = 0.03$ with PLS and $\beta = 0.22$, $p = 0.03$ with SEM). Therefore, H5 is supported, suggesting that quality of communication positively affects knowledge coordination behaviors in OSS project teams. Several prior studies have similar findings. For example, Ko et al. (2005) studied knowledge coordination between ERP implementing firms and consulting firms. They found that communication competence of the knowledge source affects knowledge coordination between these two organizations. In addition, Joshi and Sarker (2006) studied knowledge coordination behaviors in software teams made of university students. They reported that information from more reliable source is more likely to be accepted by others.

Hypothesis 6 (H6)

H6 states that knowledge coordination within an OSS team is positively associated with the team's technical achievement because teams with effective knowledge coordination need less planning ahead of time, incur fewer misunderstandings and less confusion while performing, and the members cooperate smoothly with each other (Liang et al., 1995). This hypothesis is tested by evaluating the statistical significance of the path corresponding H6. The dependent variable is technical achievement. It is measured in terms of the percentage of project issues (e.g., bugs, feature requests, and patches) resolved. The independent variable is knowledge coordination. It is measured using the survey answers from project administrators. Table 32 provides the results.

Table 32. Hypothesis Testing – H6

Hypothesis	Testing Method	Path Coefficient	Standard Error	T Statistics	P-value (one-tail)
H6	PLS	-0.13	0.11	1.14	0.13
	SEM	-0.19	2.93	1.55	0.06

The results from both testing methods indicate that the path is negative rather than the hypothesized positive ($\beta = -0.13$, $p = 0.13$ with PLS and $\beta = -0.19$, $p = 0.06$ with SEM). Therefore, H6 is not supported.

Several prior studies examined relations similar to those in H6. Their results are not consistent with the findings reported here. For example, Faraj and Sproull (2000) studied expertise coordination in software development teams and found that expertise coordination has positive effects on both team effectiveness and efficiency. Kanawattanachai and Yoo (2007) studied knowledge coordination in virtual teams of MBA students, and reported that knowledge coordination positively affects team performance.

This inconsistency might be explained by opportunity costs associated with knowledge coordination (Haas & Hansen, 2005). In a typical knowledge coordination activity, knowledge is transferred from a source to a recipient (Joshi et al., 2005), for instance, from a seasoned developer to an inexperienced developer. The inexperience developer, as the recipient, can benefit much from the activity, such as learning best practices and possible solutions for problems on hand. However, the experienced developer, as the source, incurs opportunity costs: the effort and time expended in transferring knowledge (Haas & Hansen, 2005). Thus, they could have spent time and effort on project development issues, such as, fixing a bug, devising a new feature, or coding a patch. Therefore, the team, as a whole, might not benefit from knowledge coordination if learning benefits cancel out opportunity costs incurred. This is probably the reason why data in this study shows that there is no relation between knowledge coordination and technical achievement of the team. Even worse, the team, as a whole, might perform poorer if opportunity costs exceed learning benefits produced by knowledge coordination.

Hypothesis 7 (H7)

H7 states that the quality of communication among the members of an OSS team is positively associated with the team's technical achievement. This hypothesis is based on the reasoning that timely and relevant communication enables the developers to make informed design decisions and amass necessary information to solve emerging problems. This type of communication is especially significant for the OSS teams (Stewart & Gosain, 2006) since their members rarely, if ever, meet face-to-face. The hypothesis is tested by evaluating the statistical significance of the path corresponding H7. The dependent variable is technical achievement, and is measured in terms of the percentage of project issues resolved. The independent variable is communication quality, and is measured using the survey answers from project administrators. Table 33 provides the results.

Table 33. Hypothesis Testing – H7

Hypothesis	Testing Method	Path Coefficient	Standard Error	T Statistics	P-value (one-tail)
H7	PLS	0.34	0.10	3.44	0.00
	SEM	0.40	2.52	3.09	0.00

The results from both testing methods show that the relation between communication quality and technical achievement is positive and statistically significant (β = 0.34, p = 0.00 with PLS and β = 0.40, p = 0.00 with SEM). Therefore, H7 is supported. This result is consistent with the finding reported by Stewart and Gosain (2006). They also found a positive relation between communication quality and performance of OSS project teams.

Chapter Summary

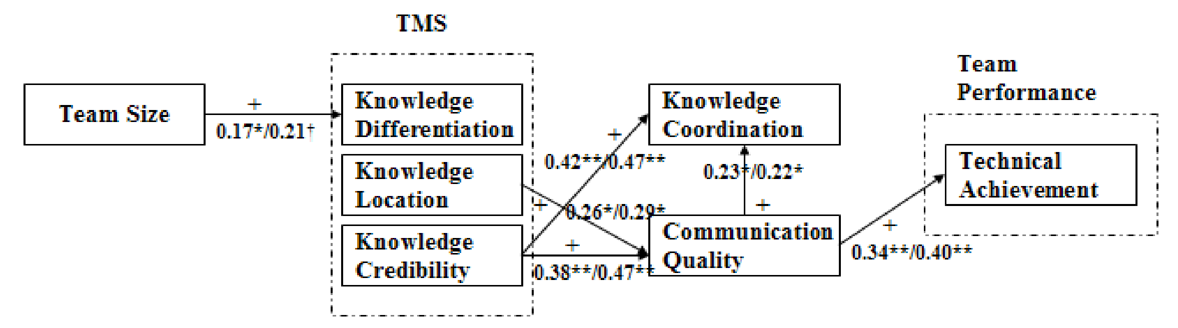
Table 34 summarizes the results of hypothesis testing. The study hypothesized two antecedents (i.e., communication volume and team size) for the TMS development. However, the results demonstrate communications taking place in the developer-mailing lists and through commit logs are not important for any dimensions of TMS development in virtual teams like OSS project teams. Team size has positive impacts on knowledge differentiation but does not affect knowledge location and knowledge credibility.

The results also validate the importance of TMS for knowledge coordination behaviors and communication quality among the OSS developers. Specifically, knowledge credibility has positive impacts on knowledge coordination and communication quality, and knowledge location positively affects communication quality. Furthermore, communication quality shows the positive influence on knowledge coordination and team performance.

Table 34. Summary of Hypotheses Testing (Phase II)

Hypothesis	Results
H1a. Communication Volume → (+)Knowledge Differentiation	Not Supported
H1b. Communication Volume → (+)Knowledge Location	Not Supported
H1c. Communication Volume → (+)Knowledge Credibility	Not Supported
H2a. Team Size → (-)Knowledge Differentiation	Not Supported* (* = significant but opposite sign)
H2b. Team Size → (-)Knowledge Location	Not Supported
H2c. Team Size → (-)Knowledge Credibility	Not Supported
H3a. Knowledge Differentiation → (+)Knowledge Coordination	Not Supported
H3b. Knowledge Location → (+)Knowledge Coordination	Not Supported
H3c. Knowledge Credibility → (+)Knowledge Coordination	Supported
H4a. Knowledge Differentiation → (+)Communication Quality	Not Supported
H4b. Knowledge Location → (+)Communication Quality	Supported
H4c. Knowledge Credibility → (+)Communication Quality	Supported
H5. Communication Quality → (+)Knowledge Coordination	Supported
H6. Knowledge Coordination → (+)Technical Achievement	Not Supported
H7. Communication Quality → (+)Technical Achievement	Supported

Figure 8 blow shows the significant paths of the research model in Phase II. The numbers near each path indicate the path coefficients and significant levels reported in this study. The first number is derived from PLS, and the second number is derived from SEM.



**p ≤ 0.00 one-tail; *p ≤ 0.05 one-tail; † ≤ 0.10 one-tail

Figure 8. Significant Paths of the Research Model (Phase II)

CHAPTER SIX: CONTRIBUTIONS, LIMITATIONS, AND FUTURE RESERCH DIRECTIONS

As Peter Drucker (1999) pointed out, "the most valuable asset of a 21st-centry institution (whether business or nonbusiness) will be its knowledge workers and their productivity" (p. 79). This statement is especially true for software development teams given the knowledge intensive nature of their "business" of producing information products. Even though the software market primarily consists of proprietary software companies, OSS has gained momentum in recent years. Under a seemingly bizarre setting (e.g., no formal organization and the volunteer nature of the workforce), OSS developers, who may be globally dispersed, work together to develop quality software. However, it is largely unknown how they have managed to achieve this end. Therefore, this dissertation set out to answer the question: ***How do the members of an OSS project team coordinate their knowledge of different domains to bear on software development tasks?***

Drawn on OSS literature, knowledge management research, and the TMS theory, a research model was proposed in an attempt to explore knowledge coordination mechanisms in OSS project teams. The model mainly addresses the relations among TMS, knowledge coordination, and communication quality, and was largely confirmed by surveying 95 OSS project teams. The results demonstrated the importance of TMS for knowledge coordination behaviors and communication quality of the OSS developers. Furthermore, communication quality shows a positive influence on team performance. These results contribute to the current literature as well as management practice.

Contributions to Literature

This dissertation specifically contributes to the TMS, knowledge management, and the OSS literature in the following aspects. Although there is general agreement that TMS is a multidimensional construct, most of the prior research bundled the dimensions together and studied TMS as a single second-order construct, under the assumption that different TMS dimensions develop and subsequently affect team behaviors in a homogenous fashion. However, this study followed the recommendation from Kanawattanachai and Yoo (2007) and separately examined the TMS dimensions. Drawn on previous literature and Wegner's original theorization of TMS, this research specifically looked into three dimensions of TMS: knowledge differentiation, knowledge location, and knowledge credibility. The results demonstrate that the "homogeneity" assumption is questionable at least in the OSS setting. Among the three dimensions examined, team size effects knowledge differentiation, but does not impact the other two dimensions. In terms of their influence on team behaviors, knowledge credibility facilitates knowledge coordination, and knowledge location and credibility have positive impacts on communication quality; however, knowledge differentiation does not promote either knowledge coordination or communication quality. Therefore, the results from this study indicate that the development of the TMS dimensions and their impacts on team behaviors can be indeed "heterogeneous," and inform researchers in the field the necessity of studying the TMS dimensions distinctly.

Despite the large number of TMS studies, research on TMS in virtual teams remains scant. Only a handful of such studies are identified in current literature. Therefore, this dissertation contributes to the literature by studying TMS in virtual teams such as OSS project teams. By surveying a large number of OSS project administrators, it is established that TMS indeed can develop in a virtual setting such as the OSS project teams. Knowledge location and credibility have significant relations with knowledge coordination and communication quality among the OSS developers. These results corroborate the similar finding of Kanawattanachai and Yoo (2007), and thus refute Lewis's claim that virtual teams might not be able to develop the effective TMS (2004).

This research is also one of the first studies looking into the knowledge coordination issue in the unique OSS setting, where the major workforce consists of voluntary developers and communication relies heavily on lean media. Hence, it contributes to the theoretical advancement in the area of knowledge management in general and OSS in particular. Specifically, the research examined the relations among TMS, communication quality, and knowledge coordination. The results show that two TMS dimensions, knowledge location and credibility, can promote timely and relevant communication. This type of communication, in turn, helps OSS developers to integrate their expertise to bear on software development tasks. Moreover, the results suggest that knowledge credibility has a direct positive bearing on knowledge coordination behaviors of the OSS developers.

Finally, the survey scales for TMS, knowledge coordination, and communication quality were developed and validated through one conceptual validation and two field tests. The wording of the scales is fairly generic rather than task specific. Therefore, they can be used to study other virtual teams with minimal modification, and the results from different studies can be easily compared and contrasted. This will facilitate accumulation of knowledge regarding the phenomenon researched.

Contributions to Practice

This dissertation developed and validated a TMS model showing, in the OSS setting, how different TMS dimensions impact knowledge coordination, communication quality, and ultimately team performance. The model can offer practical suggestions to OSS project administrators in the following aspects. Project administrators can improve communication quality among developers by helping them become familiar with each other's expertise. This can be done by implementing a knowledge map or directory on the project website, indicating each member's areas of expertise. Organizations, such as IBM, have used such a directory in its internal knowledge portal, serving as a way to help knowledge workers familiarize themselves with their colleagues (Mack, Ravin, & Byrd, 2001). Moreover, project administrators should attempt to raise the level of confidence that developers have in their team members' expertise because it can facilitate knowledge coordination and contribute to communication quality.

Project administrators can use the survey scales developed in this study to gather information about TMS, knowledge coordination, and communication quality in their software development teams. The information can help management diagnose the potential problems the teams are experiencing.

In addition to managerial suggestions to OSS project administrators in particular discussed above, the results of this study have implications for organizations experimenting with the open source mode of software development (Stewart & Gosain, 2006). A number of commercial companies have been doing so. Examples include Microsoft's CodePlex website, Sun Microsystems' OpenOffice.org site, and Google's Google code website. This study suggests that the companies wishing to leverage OSS project teams should focus their resources on facilitating the TMS development within the teams, especially the knowledge location and knowledge credibility dimensions. This is because these two dimensions were reported to have positive effects on communication quality, which, in turn, improves team performance.

Limitations

The results of the study have to be interpreted within the context of its limitations. The sample was exclusively drawn from two out of fourteen project categories on the Sourceforge website. As discussed in Chapter Four, limiting the study to two project categories was intended to minimize the chance of confounding variables, which might be introduced by category differences. However, this also limits the generalizability of the study. It remains to be tested whether the findings reported in this study are applicable to other categories of projects.

Another related issue is the power. Ninety-five projects were surveyed in Phase II. Although the sample size is well beyond the sample size recommended for PLS (Chin, 1998), it is below the minimum sample size required (i.e., 100 observations) for SEM (Ding et al., 1995). A post hoc power analysis was conducted to compute the power achieved by the statistical analysis discussed in Chapter Five. The power is the probability that a statistical test leads to rejecting a false null hypothesis (Cohen, 1988). It is equal to 1- β, where β represents the type II error. In the post hoc analysis, the power is a function of the given α (i.e., the type I error), sample size, and population effect size (Faul, Lang, & Buchner, 2007). Phase II has α of 0.05 and the sample size of 95 projects. The population effect size, f^2, measures the magnitude of the effect that a hypothesized independent variable has on a dependent variable in the population. It can be calculated using the general formula (Cohen, 1988): PV_S(population)/PV_E(population). PV_S refers to the portion of variance in a dependent variable that is explained by a hypothesized independent variable in the population. PV_E, on the other hand, refers to the portion of variance in the dependent variable that is not explained by the same independent in the population. However, no prior research suggests the values of PV_S and PV_E for each relation examined in this study. Therefore, the population effect sizes of these relations are unknown. Alternatively, Faul et al. (2007) suggested that researchers can use Cohen's definitions (1988) of small (i.e., $f^2 = 0.02$), medium (i.e., $f^2 = 0.15$), and large ($f^2 = 0.35$) for the population effect size specification under such circumstance. Computed power for this study, based on Cohen's (1998) definitions of effect size, is shown in Table 35.

Table 35. Results of Power Analysis

Effect Size (f^2)	Type I Error (α)	Sample Size	Power (1-β)
Small (f^2 = 0.02)	α = 0.5, one-tail	95	0.39
Medium (f^2 = 0.15)	α = 0.5, one-tail	95	0.98
Large (f^2 = 0.35).	α = 0.5, one-tail	95	1.00

A power of 0.80 is generally considered sufficient (Cohen, 1988). As shown, if the effect size effect is medium or large, the statistical test conducted in Chapter Five had more than 98% probability of finding a significant relation when the relation truly exists. However, if the effect size is small, the statistical test only had 39% probability of finding a significant relation when the relation truly exists.

Directions for Future Research

This study asked project administrators to assess their project teams' TMS, knowledge coordination, and communication quality. Even though this key informant approach is a common practice for studying team-level constructs (Sparrowe, Liden, Wayne, & Kraimer, 2001; Stewart & Gosain, 2006), some researchers suggest alternative approaches, such as aggregation of the individual team members' assessments (e.g., Fuller, Hardin, & Davison, 2007) and the consensus through the members' discussion (e.g., Gibson, Randel, & Earley, 2000; Hardin, Fuller, & Valachich, 2006). Therefore, future research might adopt these alternative approaches to re-examine the research model proposed in this study. The results derived from different measurement approaches might then be compared and contrasted to identify which approach is more appropriate for studying team-level constructs in the OSS setting.

This study has largely demonstrated the importance of TMS in OSS project teams. Nevertheless, it does not provide much insight on what factors influence the development of TMS. Two antecedents (i.e., communication volume and team size) were proposed, but the results show that they have very little impact on TMS. Hence, an important direction for future study is to focus on the development process of TMS, that is, how TMS develops in the first place. Some prior research (e.g., Kanawattanachai & Yoo, 2006; Moreland, 1999; Palazzolo et al., 2006) suggested that factors, such as cultural diversity, shared task experience, and familiarity among team members, might affect the development process. Future research should certainly explore the relations between these factors and TMS in the context of OSS teams.

Several prior studies (e.g., Faraj & Sproull, 2000; Tiwana, 2004) have established a positive relation between knowledge coordination and team performance. However, the results of this study show that knowledge coordination behaviors of the OSS developers do not necessarily lead to improved performance of their teams. This may partially be due to possible opportunity costs (Haas & Hansen, 2005). The costs are incurred when knowledge sources (e.g., experienced OSS developers) expend time and effort to transfer their knowledge to recipients (e.g., inexperienced OSS developers). Therefore, future research should pay attention to opportunity costs associated with knowledge coordination rather than just benefits.

Summary

The purpose of this research is to investigate the mechanism of knowledge coordination within OSS project teams. The knowledge intensive nature of software development (Faraj & Sproull, 2000; Tiwana, 2004) dictates that knowledge coordination is the key factor affecting performance of OSS project teams. From the TMS perspective, this research has developed and verified a framework, which explores the relations among TMS, knowledge coordination, communication quality, and team performance.   This framework has managerial implications for OSS project administrators as well as corporations wishing to adopt OSS work practices. It is hoped that the findings reported here inspire researchers to further explore the OSS phenomenon and particularly the knowledge management aspect of this phenomenon in order to “improve the effectiveness of software engineering as a human and team practice” (Crowston et al., 2004, p. 18).

APPENDICES

APPENDIX A

Dear OSS Developers:

Thanks for helping with this survey on OSS project teams.

This brief survey will only take less than five minutes to fill out.
It will help us immensely in understanding the dynamics in OSS project teams.

To begin, place the access code we sent to you via e-mail in the box below.

Next, click the NEXT button below to go to the first question of the survey.

Please tell us of which OSS project you are the project administrator/manager.
If you are involved in more than one project, report the name of the project you are most involved with.

Please consider your experiences particularly in the project you just reported when answering the following questions.

Think about the communication among members on your project team. Please use the following scale to rate how frequently each kind of communication listed below occurs: 1 = never; 2 = very infrequently; 3 = infrequently; 4 = sometimes; 5 = frequently; 6 = very frequently; 7 = all the time

1. Members on our team answer each other's questions in a timely manner.

2. Our team members' responses to each other's questions are correct and useful.

3. Members on our team answer each other's questions in a thoughtful manner.

4. Our team members' responses to each other's questions are irrelevant.

5. Members on our team answer each other's questions after a long delay.

The specialization of knowledge that your team possesses is represented by each of the following statements. Please indicate the extent to which you agree with each statement about your team using the following scale: 1 = strongly disagree; 2 = disagree; 3 = disagree somewhat; 4 = neither agree nor disagree; 5 = agree somewhat; 6 = agree; 7 = strongly agree

1. Each team member has specialized knowledge of some aspect of our project.	
2. Different team members are responsible for different domains of expertise needed for our project.	
3. Each team member has knowledge about some aspect of our project that no other team member on the team has.	
4. The specialized knowledge of several different members is needed to complete our project.	
5. Members of our team specialize in different aspects of the project.	
6. Members on our team have project-relevant knowledge that overlaps each other.	
7. Members on our team are "generalists."	

The location of knowledge that your team possesses is represented by each of the following statements. Please indicate the extent to which you agree with each statement about your team using the following scale: 1 = strongly disagree; 2 = disagree; 3 = disagree somewhat; 4 = neither agree nor disagree; 5 = agree somewhat; 6 = agree; 7 = strongly agree

Our team has a good "map" of each member's talents and skills.	
2. Members on our team either volunteer for or are assigned to tasks commensurate with their task-relevant knowledge and skills.	
3. Members on our team know what task-related skills and knowledge they each possess.	
4. Members on our team know who has specialized skills and knowledge that is relevant to their work.	
5. If one member has a question about some aspect of our project, this member knows who on the team she or he should ask for the answer.	
6. Our members have a hard time identifying the experts on the team.	
7. Our members have no idea what special knowledge and expertise other members on the team possess.	

The credibility of knowledge that your team possesses is represented by each of the following statements. Please indicate the extent to which you agree with each statement about your team using the following scale: 1 = strongly disagree; 2 = disagree; 3 = disagree somewhat; 4 = neither agree nor disagree; 5 = agree somewhat; 6 = agree; 7 = strongly agree

1. The members on our team do not have doubts on project-relevant suggestions from other members.	
2. The members on our team trust that the other members' knowledge about the project is credible.	

3. The members on our team are confident when applying the knowledge provided by other members to the project tasks at hand.

4. The members on our team did not have much faith in the other members' "expertise."

5. The members on our team like to double-check the knowledge provided by other members before applying it to the project tasks at hand.

Each of the statement below refers to how the members on your project help each other on project tasks. Please indicate the extent to which you agree or disagree with each statement about your team using the following scale: 1 = strongly disagree; 2 = disagree; 3 = disagree somewhat; 4 = neither agree nor disagree; 5 = agree somewhat; 6 = agree; 7 = strongly agree

1. Members in our team share their special knowledge and expertise with one another.

2. If someone in our team has some special knowledge about how to perform the project task, he or she is not likely to tell the other member about it.

3. Members in our team virtually do not share their information, knowledge, or skills with one another.

4. More knowledgeable members in our team willingly make their knowledge and expertise available to other members.

5. Project tasks are completed by integrating the specialized knowledge of different members in our team.

Do you have any suggestions for improving this survey?

Please click DONE button below to submit your answers.
Thanks for your input and time.

APPENDIX B

Dear OSS Developers:

Thanks for helping with this survey on OSS project teams.

This brief survey will only take less than five minutes to fill out.
It will help us immensely in understanding the dynamics in OSS project teams.

To begin, place the access code we sent to you via e-mail in the box below.

Next, click the NEXT button below to go to the first question of the survey.

Please tell us of which OSS project you are the project administrator/manager.
If you are involved in more than one project, report the name of the project you are most involved with.

Please consider your experiences particularly in the project you just reported when answering the following questions.

Think about the communication among members on your project team. Please use the following scale to rate how frequently each kind of communication listed below occurs: 1 = never; 2 = very infrequently; 3 = infrequently; 4 = sometimes; 5 = frequently; 6 = very frequently; 7 = all the time

1. Members on our team answer each other's questions in a timely manner.
2. Our team members' responses to each other's questions are correct and useful.
3. Members on our team answer each other's questions in a thoughtful manner.

The specialization of knowledge that your team possesses is represented by each of the following statements. Please indicate the extent to which you agree with each statement about your team using the following scale: 1 = strongly disagree; 2 = disagree; 3 = disagree somewhat; 4 = neither agree nor disagree; 5 = agree somewhat; 6 = agree; 7 = strongly agree

1. Each team member has specialized knowledge of some aspect of our project.
2. Different team members are responsible for different domains of expertise needed for our project.
3. Each team member has knowledge about some aspect of our project that no other team member on the team has.

4. The specialized knowledge of several different members is needed to complete our project.

The location of knowledge that your team possesses is represented by each of the following statements. Please indicate the extent to which you agree with each statement about your team using the following scale: 1 = strongly disagree; 2 = disagree; 3 = disagree somewhat; 4 = neither agree nor disagree; 5 = agree somewhat; 6 = agree; 7 = strongly agree

1. Our team has a good "map" of each member's talents and skills.

2. Members on our team know what task-related skills and knowledge they each possess.

3. Members on our team know who has specialized skills and knowledge that is relevant to their work.

4. If one member has a question about some aspect of our project, this member knows who on the team she or he should ask for the answer.

5. Our members have a hard time identifying the experts on the team.

6. Our members have no idea what special knowledge and expertise other members on the team possess.

The credibility of knowledge that your team possesses is represented by each of the following statements. Please indicate the extent to which you agree with each statement about your team using the following scale: 1 = strongly disagree; 2 = disagree; 3 = disagree somewhat; 4 = neither agree nor disagree; 5 = agree somewhat; 6 = agree; 7 = strongly agree

1. The members on our team trust that the other members' knowledge about the project is credible.

2. The members on our team are confident when applying the knowledge provided by other members to the project tasks at hand.

3. The members on our team did not have much faith in the other members' "expertise."

Each of the statement below refers to how the members on your project help each other on project tasks. Please indicate the extent to which you agree or disagree with each statement about your team using the following scale: 1 = strongly disagree; 2 = disagree; 3 = disagree somewhat; 4 = neither agree nor disagree; 5 = agree somewhat; 6 = agree; 7 = strongly agree

1. Members in our team share their special knowledge and expertise with one another.

2. Members in our team virtually do not share their information, knowledge, or skills with one another.

3. More knowledgeable members in our team willingly make their knowledge and expertise available to other members.

APPENDIX C

Pre-notice Email

A few days from now you will receive a request at this same e-mail address to fill out a brief web survey, which concerns your experiences in the SourceForge community and how you feel about the project(s) that your are involved with there.

The information gathered through the survey will be used for a research project being conducted in The University of Texas at San Antonio (UTSA). This research will help software development researchers as well as practitioners understand the dynamics in open source software (OSS) project teams, and whether the practices in the OSS community are applicable to software development in general.

If you have any questions, feel free to contact me (Trent Chen) at (001)210-254-3512, or by e-mail at trent.chen@utsa.edu.

Thank you for time and consideration. It's only with the generous help of people like you that our research can be successful.

Sincerely,

Trent Chen
The Department of Information Systems and Technology Management
UTSA

P.S. We will be enclosing a token of appreciation with questionnaire as a way of saying thanks.

Second Email

Title of Project: Transactive Memory Systems in Open Source Software Project Teams: An Examination of Formation of Open Source Software Teams' Transactive Memory Systems and Its Impacts on Team Performance
Principal Investigator: Xiaogang (Trent) Chen

The enclosed link will lead you to the brief web survey regarding your experiences in the SourceForge community which I (Trent Chen) notified you about via email a few days ago. This survey is part of a study being conducted in the University of Texas at San Antonio (UTSA) to learn how the members on an Open Source Software (OSS) team work together to achieve their project goals. We are asking you to take part in this study because you are the project administrator/manager and are most familiar with the team's internal dynamics, activities, and accomplishments. 300 project administrators/managers are expected to participate in this study.

If you agree to participate, you will respond to an online survey about your experiences in the OSS team. It takes less than five minutes to complete the survey. Your answers are completely confidential and will be released only as summaries in which no individual's answers can be identified. No risks or discomforts are anticipated from this study.

As a way of saying thanks for your help, you are offered a chance to win a $200 lottery. The lottery will be dispersed one week after this e-mail through the electronic donation system in SourceForge.

Your participation in this study is entirely voluntary. You are free to skip questions you find objectionable in the survey and withdraw from the study at any time. However, you can help us very much by taking a few minutes to share your experiences and opinions about the OSS project teams. If you want to participate, your answering the survey will indicate you are giving consent.

If you have any questions or comments about this study, you can reach me at (001)210- 254-3512, or by e-mail at trent.chen@utsa.edu. If you have questions about your rights as a research subject you can contact the University of Texas Institutional Review Board at (001)210-458-6473, or you can email IRB@utsa.edu.

To begin, click the link (http://www.surveymonkey.com/s.aspx?sm=EnFaxVv4avveV5PGITbBUg_3d_3d), and then you will be asked for the access code (your access code is: .). Next, click NEXT button to go to the first question of the survey. You will be then asked a series of questions regarding your experiences particularly in the project.

Sincerely,
Trent Chen
The Department of Information Systems and Technology Management
UTSA

Third Email

About a few days ago we sent you an online survey link via e-mail to seek your opinions about open source software (OSS) projects at the SourceForge site. As of today, we have not received a completed survey from you. We are contacting you now in hopes of obtaining the insights only OSS project administrators/managers like you can provide, and which other OSS developers can potentially benefit from.

As a way of saying thanks for your help, you are offered a chance to win a $200 lottery. The lottery will be dispersed a few days after this e-mail through the electronic donation system in SourceForge.

As we mentioned before, your answers are confidential and will be combined with others before providing results to the public. In case the previous e-mail has been deleted from your e-mail box, we have included the online survey link and your access code again.

Your access code:

The online survey link:
http://www.surveymonkey.com/s.aspx?sm=EnFaxVv4avveV5PGITbBUg_3d_3d

Should you have any questions or concerns, feel free to contact me (Trent Chen) at (001)210-254-3512, or by e-mail at trent.chen@utsa.edu. Thank you for your cooperation.

Trent Chen
The Department of Information Systems and Technology Management
UTSA

APPENDIX D

Table D1. ANOVA Results

		Sum of Squares	df	Mean Square	F	Sig.
Technical Achievement	Between Groups	864.110	1	864.110	1.939	.167
	Within Groups	41443.754	93	445.632		
	Total	42307.864	94			
CQ1	Between Groups	.866	1	.866	.480	.490
	Within Groups	167.934	93	1.806		
	Total	168.800	94			
CQ2	Between Groups	1.616	1	1.616	1.268	.263
	Within Groups	118.532	93	1.275		
	Total	120.147	94			
CQ3	Between Groups	3.228	1	3.228	2.235	.138
	Within Groups	134.309	93	1.444		
	Total	137.537	94			
KD1	Between Groups	.177	1	.177	.119	.731
	Within Groups	137.970	93	1.484		
	Total	138.147	94			
KD2	Between Groups	.267	1	.267	.113	.738
	Within Groups	219.691	93	2.362		
	Total	219.958	94			
KD3	Between Groups	.103	1	.103	.037	.848
	Within Groups	257.729	93	2.771		
	Total	257.832	94			

Table D1_continued

KD4	Between Groups	4.305	1	4.305	1.269	.263
	Within Groups	315.443	93	3.392		
	Total	319.747	94			
KL1	Between Groups	.086	1	.086	.047	.829
	Within Groups	171.661	93	1.846		
	Total	171.747	94			
KL3	Between Groups	.127	1	.127	.084	.773
	Within Groups	140.505	93	1.511		
	Total	140.632	94			
KL4	Between Groups	.011	1	.011	.009	.925
	Within Groups	113.526	93	1.221		
	Total	113.537	94			
KL5	Between Groups	.001	1	.001	.001	.975
	Within Groups	109.157	93	1.174		
	Total	109.158	94			
KL6	Between Groups	.504	1	.504	.322	.572
	Within Groups	145.854	93	1.568		
	Total	146.358	94			
KL7	Between Groups	.000	1	.000	.000	.995
	Within Groups	174.358	93	1.875		
	Total	174.358	94			
KCR2	Between Groups	1.467	1	1.467	2.126	.148
	Within Groups	64.154	93	.690		
	Total	65.621	94			

Table D1_continued

KCR3	Between Groups	1.862	1	1.862	2.600	.110
	Within Groups	66.622	93	.716		
	Total	68.484	94			
KCR4	Between Groups	.817	1	.817	.917	.341
	Within Groups	82.804	93	.890		
	Total	83.621	94			
KCO1	Between Groups	.576	1	.576	.561	.456
	Within Groups	95.571	93	1.028		
	Total	96.147	94			
KCO3	Between Groups	.563	1	.563	.429	.514
	Within Groups	121.921	93	1.311		
	Total	122.484	94			
KCO4	Between Groups	.190	1	.190	.154	.696
	Within Groups	115.115	93	1.238		
	Total	115.305	94			

APPENDIX E

UTSA The University of Texas at San Antonio
Office of Research Integrity and Compliance

FWA #00003861

October 16, 2008

Xiaogang (Trent) Chen
11600 Huebner Rd., Apt #2516
San Antonio, TX 78230

NOTICE OF REAPPROVAL

Dear Mr. Chen:

Re: IRB #08-015
Title: "Transactive Memory Systems in Open Source Software Project Teams: An Examination of Formation of Open Source Software Teams' Transactive Memory Systems and Its Impacts on Team Performance"
Site(s): UTSA

In accordance with Federal regulations for continuing review, the above referenced protocol was reviewed on October 16, 2008, by Expedited Review on behalf of the Institutional Review Board and was approved under the following Expedited Review Category:

(7) Research on individual or group characteristics or behavior (including, but not limited to, research on perception, cognition, motivation, identity, language, communication, cultural beliefs or practices, and social behavior) or research employing survey, interview, oral history, focus group, program evaluation, human factors evaluation, or quality assurance methodologies. (NOTE: Some research in this category may be exempt from the HHS regulations for the protection of human subjects. 45 CFR 46.101(b)(2) and (b)(3). This listing refers only to research that is not exempt.)

The study is open to enrollment. Data are being analyzed.

The protocol continues to meet the criteria and is approved for a waiver of documentation of informed consent under 45 CFR 46.117(c)(2). The research presents no more than minimal risk of harm to subjects and involves no procedures for which written consent is normally required outside the research context.

One consent form was approved and is stamped with the current approval period.

It was determined that the research continues to pose minimal risk to the participating subjects, and that continuing review will occur annually. The determinations for approval have been satisfied. This action will be reported to the Institutional Review Board at its next convened meeting.

RESPONSIBILITIES OF PRINCIPAL INVESTIGATOR FOR ONGOING PROTOCOLS:

1. Conduct the study only according to the protocol approved by the IRB; promptly report any protocol, policy, or regulatory deviations or violations to the IRB;

One UTSA Circle • San Antonio, Texas 78249 • (210) 458-4601 • (210) 458-5196 fax

BIBLIOGRAPHY

Anderson, J. C., & Gerbing, D. W. (1988). Structural equation modeling in practice: A review and recommended two-step approach. *Psychological Bulletin, 103*(3), 411-427.

Agarwal, R., & Karahanna, E. (2000). Time flies when you are having fun: Cognitive absorption and beliefs about information technology usage. *MIS Quarterly, 24*(4), 665-694.

Akgün, A. E., Byrne, J. C., Keskin, H., & Lynn, G. S. (2006). Transactive memory system in new product development teams. *IEEE Transactions on Engineering Management, 53*(1), 95-111.

Asay, M. (2008). NYSE banks on Red Hat. Retrieved August 3, 2008, from http://news.cnet.com/8301-13505_3-9942680-16.html.

Austin, J. R. (2003). Transactive memory in organizational groups: The effects of content, consensus, specialization, and accuracy on group performance. *Journal of Applied Psychology, 88*(5), 866-878.

Au, Y. A., Carpenter, D., Chen, X., & Clark, J. G. (2009). Virtual organizational learning in open source software development projects. *Information & Management*, *46*(1), 9-15.

Brooks, F. (1975). *The mythical man-month: Essays on software engineering*: Addison-Wesley Professional.

Carmel, E. (1999). *Global software teams: Collaboration across borders and time zones*. Upper Saddle River, NJ: Prentice Hall.

Chin, W. W. (1998). *The partial least squares approach to structural equation modeling*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Chou, C. P., & Bentler, P. M. (1995). Estimates and tests in structural equation modeling. In R. H. Hoyle (Ed.), *Structural equation modeling: Issues and applications* (pp. 37-55). Thousand Oaks, CA: Sage.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd.). Hillsdale, NJ: Lawrence Erlbaum Associates.

Cox, A. (1998). Cathedrals, bazaars and the town council. Retrieved April 13, 2007: http://www.linux-france.org/article/these/conseil-municipal/bazaar.html

Crano, W. D., & Brewer, M. B. (2002). *Principles and methods of social research*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Crowston, K., Annabi, H., Howison, J., & Masango, C. (2004). *Effective work practices for software engineering: Free/Libre open source software development*. Paper presented at the 2004 ACM Workshop on Interdisciplinary Software Engineering Research.

Crowston, K., & Howison, J. (2006). Hierarchy and centralization in free and open source software team communications. *Knowledge, Technology, & Policy, 18*(4), 65-85.

Cummings, J. N. (2004). Work Groups, structural diversity, and knowledge sharing in a global organization. *Management Science, 50*(3), 352-364.

Cusumano, M. (2005). Foreword. In J. Feller, B. Fitzgerald, S. A. Hissam, & K. R. Lakhani (Ed.), *Perspectives on Free and Open Source Software* (pp. xi-xiii). London: The MIT Press.

Damian, D., Lanubile, F., & Oppenheimer, H. L. (2003). *Addressing the challenges of software industry globalization: The workshop on global software development.* Paper presented at the 25th International Conference on Software Engineering, Portland, Oregon, USA.

DeVellis, R. F. (2003). *Scale development: Theory and applications* (2nd ed. Vol. 26). London: SAGE Publications.

Ding, L., Velicer, W. F. , & Harlow, L. L. (1995). Effects on estimation methods, number of indicators per factor, and improper solutions on structural equation modeling fit indices. *Structural Equation Modeling*, *2*, 119-143。

Drucker, P. F. (1999). Knowledge-worker productivity: The biggest challenge. *California Management Review, 41*(2), 79-94.

Erdfelder, E., Faul, F., & Buchner, A. (1996). GPOWER: A general power analysis program. *Behavior Research Methods, Instruments, & Computers*, *28*, 1-11.

Efron, B., & Tibshirani, R. J. (1998). An introduction to the bootstrap. Florida: Chapman & Hall

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

Faul, F., Erdfelder, E., Lang, A.-G., & Buchner, A. (2007). G*Power 3: A flexible statistical power analysis for the social, behavioral, and biomedical sciences. *Behavior Research Methods, 39*, 175-191.

Fornell, C., & Larcker, D. (1981). Evaluating structural equation models with unobservable variables and measurement error, *Journal of Marketing Research, 18*, 39-50.

Freedman, D., Pisani, R., & Purves, R. (1998). *Statistics* (3rd ed.). New York: W.W. Norton & Company, Inc.

Fuller, M. A., Hardin, A. M., & Davison, R. M. (2007). Efficacy in technology-mediated distributed team, *Journal of Management Information Systems*, *23*(3), 209-235.

Gallivan, M. J. (2001). Striking a balance between trust and control in a virtual organization: A content analysis of open source software case studies. *Information Systems Journal 11*(4), 277-304.

Gibson, C. B., Randel, A. E., & Earley, P. C. (2000). Understanding group efficacy: An empirical test of multiple assessment methods. *Group & Organization Management, 25*(1), 67-97.

Goodhue, D., Lewis, W., & Thompson, R. (2006). *PLS, small sample size, and statistical power in MIS research.* Paper presented at the 39th Hawaii International Conference on System Sciences, Kauai, HI.

Ghosh, R. A. (2002). *Free/libre and open source software: Survey and study*. Paper presented on the FLOSS Workshop on Advancing the Research Agenda on Free/Open Source Software, Brussels, Belgium.

Ghosh, R., & Prakash V. V. (2000). The orbiten free software survey. *Firstmonday, 5(7),* Retrieved September 26, 2008: http://www.firstmonday.dk/issue5_7/ghosh

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

Gutwin, C., Penner, R., & Schnerder, K. (2004). *Group awareness in distributed software development.* Paper presented at the Computer Supported Cooperative Work, Chicago, Illinois, USA.

Haas, M. R., & Hansen, M. T. (2005). When using knowledge can hurt performance: The value of organizational capabilities in a management consulting company. *Strategic Management Journal, 26*, 1-24.

Hardin, A. M., Fuller, M. A., & Valacich, J. S. (2006). Measuring group efficacy in virtual teams. *Small Group Research, 37*(1), 65-85.

He, J., Butler, B. S., & King, W. R. (2007). Team cognition: development and evolution in Software Project Teams. *Journal of Management Information Systems, 24*(2), 261-292.

Hollingshead, A. B. (1998a). Communication, learning, and retrieval in transactive memory systems. *Journal of Experimental Social Psychology, 34*(5), 423-442.

Hollingshead, A. B. (1998b). Retrieval processes in transactive memory systems. *Journal of Personality and Social Psychology, 74*(3), 659-671.

Hu, L., & Bentler, P. M. (1998). Fit indices in covariance structure modeling: Sensitivity to underparameterized model misspecification. *Psychological Methods, 3*(4), 424-453.

Hu, L., & Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling, 6*(1), 1-55.

Joshi, K. D., & Sarker, S. (2006, January). *Examining the role of knowledge, source, recipient, relational, and situational context on knowledge transfer among face-to-face ISD teams.* Paper presented at the 39th Hawaii International Conference on System Sciences, Kauai, HI, USA.

Joshi, K. D., Sarker, S., & Sarker, S. (2004, January). *Knowledge transfer among face-to-face information systems development team members: examining the role of knowledge, source, and relational context.* Paper presented at the 37th Hawaii International Conference on System Sciences, Big Island, HI, USA.

Joshi, K. D., Sarker, S., & Sarker, S. (2005, January). *The impact of knowledge, source, situational and relational context on knowledge transfer during ISD process.* Paper presented at the 38th Hawaii International Conference on System Sciences, Big Island, HI, USA.

Kanawattanachai, P., & Yoo, Y. (2007). The impact of knowledge coordination on virtual team performance over time, *MIS Quarterly, 31*(4), 783-808.

Kankanhalli, A., Tan, B. C. Y., & Wei, K.-K. (2005). Contributing knowledge to electronic knowledge repositories: An empirical investigation. *MIS Quarterly, 29*(1), 113-143.

Kelley, H. H. (1973). The processes of causal attribution. *American Psychologist, 28,* 107-128.

Kirsch, L. J., Sambamurthy, V., Ko, D.-G., & Purvis, R. L. (2002). Controlling information systems development projects: The view from the client. *Management Science, 48*(4), 484-498.

Ko, D.-G., Kirsch, L. J., & King, W. R. (2005). Antecedents of knowledge transfer from consultants to clients in enterprise system implementations. *MIS Quarterly, 29*(1), 59-85.

Kotlarsky, J., & Oshri, I. (2005). Social Ties, knowledge sharing and successful collaboration in globally distributed system development projects. *European Journal of Information Systems, 14*(1), 37-48.

Krogh, G. v., & Hippel, E. v. (2006). The promise of research on open source software. *Management Science, 52*(7), 975-983.

Krogh, G. v., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specializing in open source software innovation: A case study. *Research Policy, 32*(7), 1217-1241.

Kuk, G. (2006). Strategic interaction and knowledge sharing in the KDE developer mailing list. *Management Science, 52*(7), 1031-1042.

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

Lewis, K. (2003). Measuring transactive memory systems in the field: Scale development and validation. *Journal of Applied Psychology, 88*(4), 587-604.

Lewis, K. (2004). Knowledge and performance in knowledge-worker teams: A longitudinal study of transactive memory systems. *Management Science, 50*(11), 1519-1533.

Liang, D. W., Moreland, R. L., & Argote, L. (1995). Group versus individual training and group performance: The mediating role of transactive memory. *Personality and Social Psychology Bulletin, 21*(4), 384-393.

Louridas, P. (2006). Version control. *IEEE Software, 23*(1), 104-107.

Mack, R., Ravin, Y., & Byrd, R. J. (2001). Knowledge portals and the emerging digital knowledge workplace. *IBM Systems Journal, 40*(4), 925-955.

Michlmayr, M. (2005). *Software process maturity and the success of free software projects* (Vol. 130): IOS Press.

Mitchell, V. L. (2006). Knowledge integration and information technology project performance. *MIS Quarterly, 30*(4), 919-939.

Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2002). Two case studies of open source software development: Apache and Mozzilla. *ACM Transactions on Software Engineering and Methodology, 11*(3), 309-346.

Moon, J. Y., & Sproull, L. (2000). Essence of distributed work: The case of Linux kernel [Electronic Version]. *First Monday*, 5. Retrieved April 12, 2008 from URL: http://firstmonday.org/issues/issue5_11/moon/index.html.

Moore, G. C., & Benbasat, I. (1991). Development of an instrument to measure the perceptions of adopting an information technology innovation. *Information Systems Journal, 2*(3), 173-191.

Moreland, R. L. (1999). *Transactive memory: Learning who knows what in work groups and organizations*. Mahwah, NJ: Lawrence Erlbaum Associates.

Moreland, R. L., Argote, L., & Krishnan, R. (1996). *Socially Shared Cognition at Work: Transactive Memory and Group Performance*: Sage Publications, Inc.

Moreland, R. L., & Levine, J. M. (1992). Problem identification by groups. In S. Worchel, W. Wood, & J. Simpson (Eds.), *Group process and productivity* (pp. 17-48). Newbury Park, CA.: Sage.

Narduzzo, A., & Rossi, A. (2003). Modularity in action: GNU/Linux and free/open source software development model unleashed. Retrieved March 23, 2009, from http://opensource.mit.edu/papers/narduzzorossi.pdf.

Nevitt, J., & Hancock, G. R. (2001). Performance of bootstrapping approaches to model test statistics and parameter standard error estimation in structural equation modeling, *Structural Equation Modeling, 8*(3), 353-377.

Nunnally, J. C. (1978). *Psychometric theory*. New York, NY: McGraw-Hill.

Ojha, A. K. (2005). Impact of team demography on knowledge sharing in software project teams. *South Asian Journal of Management, 12*(3), 67-78.

Osterloh, M., & Rota, S. (2007). Open source software development: Just another case of collective invention? *Research Policy, 36,* 157-171.

Palazzolo, E. T., Serb, D. A., She, Y., Su, C., & Contractor, N. S. (2006). Coevolution of communication and knowledge networks in transactive memory systems: Using computational models for theoretical development. *Communication Theory, 16*(2), 223-250.

Raymond, E. S. (2000). The cathedral and the bazaar. Retrieved June 10, 2006, from http://www.tuxedo.org/~esr/

Richardson, J. T. E. (1996). Measures of effect size. *Behavior Research Methods, Instruments, & Computers*, *28*, 12-22.

Ringle, Wende, Will, & Hamburg, 2005, http://www.smartpls.de.

Robillard, P. N. (1999). The role of knowledge in software development. *Communications of the ACM, 42*(1), 87-92.

Rus, I., & Lindvall, M. (2002). Knowledge management in software engineering. *IEEE Software, 19*(3), 26-38.

Sabherwal, R., Jeyaraj, A., & Chowa, C. (2006). Information system success: individual and organizational determinants. *Management Science, 52*(12), 1849-1864.

Sanchez, R., & Mahoney, J. T. (1996). Modularity, flexibility, and knowledge management in product and organization design. *Strategic Management Journal, 17*, 63–76.

Sarker, S., Sarker, S., Nicholson, D. B., & Joshi, K. D. (2005). Knowledge transfer in virtual systems development teams: An exploratory study of four key enablers. *IEEE Transactions on Professional Communication, 48*(2), 201-218.

Scacchi, W. (2001, December). *Understanding the requirements for developing open source software systems.* Paper presented at the IEE Proceedings Software.

Sparrowe, R. T., Liden, R. C., Wayne, S. J., & Kraimer, M. L. (2001). Social networks and the performance of individuals and groups, *Academy of Management Journal, 44*(2), 316-325.

Stamelos, I., Angelis, L., Oikonomou, A., & Bleris, G. L. (2002). Code quality analysis in open source software development. *Information Systems Journal, 12(1)*, 43-60

Stevens, J. (1996). *Applied multivariate statistics for the social sciences*. Mahwah, NJ: Lawrence Erlbaum Associates.

Stewart, K. J., & Gosain, S. (2006). The impact of ideology on effectiveness in open source software development teams. *MIS Quarterly, 30*(2), 291-314.

Stone, E. F. (1978). *Research methods in organizational behavior*. Santa Monica, CA: Goodyear.

Tenenhaus, M., Vinzi, V. E. , Chatelin, Y-M., & Lauro, C. (2005). PLS path modeling *Computational Statistics & Data Analysis, 48*, 159-205

Tinsley, H. E. A., & Tinsley, D. J. (1987). Use of factor analysis in counseling psychology research. *Journal of Counseling Psychology, 34*(4), 414-424.

Tiwana, A. (2004). An empirical study of the effect of knowledge integration on software development performance *Information and Software Technology, 46*(13), 899-906.

Trice, H. M., & Beyer, J. M. (1993). *The cultures of work organizations*. Englewood Cliffs, NJ: Prentice Hall.

Voth, D. (2006). CodePlex: Microsoft's Latest Entry into Code Sharing. *IEEE Software, 23*(5), 96-98.

Wegner, D. M. (1987). *Transactive memory: A contemporary analysis of the group mind*. New York: Springer-Verlag.

Wegner, D. M., Erber, R., & Paula, R. (1991). Transactive memory in close relationships. *Journal of Personality and Social Psychology, 61*(6), 923-929.

Wegner, D. M., Giuliano, T., & Hertel, P. T. (1985). *Cognitive interdependence in close relationships*. New York: Springer-Verlag.

West, S. G., Finch, J. F., & Curran, P. J. (1995). Structural equation with nonnormal variables: Problems and remedies. In R. H. Hoyle (Ed.), *Structural equation modeling: Issues and applications* (pp. 56-75). Thousand Oaks, CA: Sage.

Wittenbaum, G. M., Vaughan, S. I., & Stasser, G. (1998). *Coordination in task performing groups*. New York: Plenum Press.

Yoo, Y., & Kanawattanachai, P. (2001). Developments of transactive memory systems and collective mind in virtual teams. *The International Journal of Organizational Analysis, 9*(2), 187-208.

VITA

Xiaogang Chen was born in Shannxi Province, P.R. China. He graduated from Xi'an Sixth Middle School and subsequently attended Xi'an University of Technology. He came to United States to continue his education. He first received his Master of Science in management information systems from Claremont Graduate University and later joined the University of Texas at San Antonio to pursue a doctoral degree in the field of information systems. His research interests include open source software, knowledge management, and virtual team cognition.