Knowledge Location, Differentiation, Credibility and Coordination in Open Source Software Development Teams

Xiaogang Chen
University of Texas at San Antonio, trent.chen@utsa.edu

Glenn Dietrich
University of Texas at San Antonio, glenn.dietrich@utsa.edu

Follow this and additional works at: http://aisel.aisnet.org/amcis2009

Recommended Citation
http://aisel.aisnet.org/amcis2009/389

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2009 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
Knowledge Location, Differentiation, Credibility, and Coordination in Open Source Software Development Teams

Xiaogang Chen  
The University of Texas at San Antonio  
Trent.chen@utsa.edu

Glenn Dietrich  
The University of Texas at San Antonio  
Glenn.dietrich@utsa.edu

ABSTRACT

A number of high-quality, large-scale, complex software systems, such as Linux, Apache, and Perl, have been successfully produced through the open source software (OSS) paradigm. This fact suggests that effective knowledge coordination must exist within some OSS teams. However, very few studies have attempted to explicate what these coordination mechanisms are. Therefore, this study examines how knowledge is coordinated between the members of an OSS team from the transactive memory system (TMS) perspective. Specifically, we investigate 1) the relation between TMS and the team members’ knowledge coordination behaviors, and 2) the relation between knowledge coordination and the team’s performance. By surveying 61 OSS project teams, the study validates the important role that TMS plays in OSS developers’ knowledge coordination behaviors, which, in turn, have positive influence on their projects’ technical achievement.

Keywords

Open source software (OSS), knowledge location, knowledge differentiation, knowledge credibility, knowledge coordination, transactive memory system (TMS), technical achievement.

INTRODUCTION

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 substitution for proprietary software. 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). Quite a few software producing firms, such as Red Hat, VA Software, and Mozilla have built their business models entirely on the OSS paradigm.

Most OSS is developed and maintained by teams of voluntary developers, scattered around the world (Crowston et al., 2004). These developers interact with each other almost exclusively through lean media (e.g., mailing-lists). Furthermore, most OSS 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 teams. Particularly, because of the knowledge intensive nature of software development (Robillard, 1999), some mechanisms must be employed to coordinate knowledge distributed among different members of an OSS team. However, little is known about knowledge coordination in the OSS setting (Crowston et al., 2004; Mockus et al., 2002). 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, we intend 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). 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 and seek the possibility of applying the open source modes of coordination and organization to other areas (Ghosh, 2002).
THEORETICAL BACKGROUND AND HYPOTHESES

The Transactive Memory System (TMS) theory provides a useful theoretical perspective to study knowledge coordination mechanisms in an OSS team because of its specific focus on how the different knowledge that team members possess is integrated to bear on team tasks (Lewis, 2003). Wegner 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 on each other as “external memory storage” (Wegner, 1987, p. 187) to remember some group-relevant information. While the information itself is distributed among the different group members, each member commonly shares the information about who knows what. This interdependence results in a group “knowledge-holding system that is larger and more complex than” (Wegner, 1987, p. 189) any individual member’s own memory system. Meanwhile, each member can easily access the information stored in this system because the location of information is shared among the members. Wegner termed this knowledge-holding system a 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 in the domain; (2) known to have the access to knowledge in the domain; (3) or assigned by an authority to the domain. 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. 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 might subsequently generate new knowledge that is qualitatively different from any single piece (Wenger, 1987).

Laboratory and field studies have demonstrated the importance of TMS in a variety of group settings, such as dating couples, consulting teams, and new product development teams (e.g., Austin, 2003; Hollingshead, 1998). Moreover, these studies have found that TMS is a multi-dimensional construct. For example, Moreland et al. (1999) and Lewis (2003) posit three dimensions for TMS. Based on these prior studies and original theorization of TMS, we particularly consider three dimensions of TMS in this study: knowledge location, knowledge differentiation, and knowledge credibility, and hypothesize the relations between the three dimensions and knowledge coordination behaviors of OSS team members.

Knowledge coordination

Knowledge is the most important resource for a software development team (Faraj and Sproull, 2000), yet being possessed by different team members. It must be effectively coordinated to influence the team performance (Tiwana, 2004). Knowledge coordination in this study, adapted from previous literature (e.g., Faraj and Sproull, 2000; Tiwana, 2004), refers to the extent to which the members of an OSS team integrate their different domains of expertise to bear on software development tasks.

Knowledge location

We consider knowledge location as one dimension of TMS and define 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. Wegner conceived that shared understanding among team members about who knows what (i.e., knowledge location) is the central mechanism for the team member to integrate their different knowledge together. This location information functions as an important integrative mechanism for coordination behaviors between team members (Faraj and Sproull, 2000). Recent studies show that knowledge location is especially critical for software development teams. For instance, He et al. (2007) found that with expertise location information, a team can assign its members with the tasks commensurate with their specialties and skills as well as points team members where to obtain knowledge needed when problems arise. Drawing on the above research, we hypothesize:

H1: Knowledge location is positively associated with knowledge coordination behaviors of the members of an OSS team.

Knowledge differentiation

We consider knowledge differentiation as another dimension of TMS and define knowledge differentiation as the extent to which the developers of an OSS team specialize in different knowledge domains relevant to the team project. Wegner claimed that TMS is essentially a knowledge-holding structure where diverse domains of knowledge from different team members tie together to provide a shared awareness of who knows what. Laboratory and field studies have demonstrated the importance of TMS in a variety of group settings, such as dating couples, consulting teams, and new product development teams (e.g., Austin, 2003; Hollingshead, 1998). Moreover, these studies have found that TMS is a multi-dimensional construct. For example, Moreland et al. (1999) and Lewis (2003) posit three dimensions for TMS. Based on these prior studies and original theorization of TMS, we particularly consider three dimensions of TMS in this study: knowledge location, knowledge differentiation, and knowledge credibility, and hypothesize the relations between the three dimensions and knowledge coordination behaviors of OSS team members.

Knowledge coordination

Knowledge is the most important resource for a software development team (Faraj and Sproull, 2000), yet being possessed by different team members. It must be effectively coordinated to influence the team performance (Tiwana, 2004). Knowledge coordination in this study, adapted from previous literature (e.g., Faraj and Sproull, 2000; Tiwana, 2004), refers to the extent to which the members of an OSS team integrate their different domains of expertise to bear on software development tasks.

Knowledge location

We consider knowledge location as one dimension of TMS and define 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. Wegner conceived that shared understanding among team members about who knows what (i.e., knowledge location) is the central mechanism for the team member to integrate their different knowledge together. This location information functions as an important integrative mechanism for coordination behaviors between team members (Faraj and Sproull, 2000). Recent studies show that knowledge location is especially critical for software development teams. For instance, He et al. (2007) found that with expertise location information, a team can assign its members with the tasks commensurate with their specialties and skills as well as points team members where to obtain knowledge needed when problems arise. Drawing on the above research, we hypothesize:

H1: Knowledge location is positively associated with knowledge coordination behaviors of the members of an OSS team.

Knowledge differentiation

We consider knowledge differentiation as another dimension of TMS and define knowledge differentiation as the extent to which the developers of an OSS team specialize in different knowledge domains relevant to the team project. Wegner claimed that TMS is essentially a knowledge-holding structure where diverse domains of knowledge from different team members tie together to provide a shared awareness of who knows what.
members are stored and indexed. This differentiated knowledge-holding structure starts to form as team members accept responsibility for knowledge of different domains that are relevant for team tasks (Wegner, 1987). 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 and 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 (Tiwana, 2004).

However, knowledge differentiation might cause “contribution barriers” (Krogh et al., 2003, p. 1231). An OSS project typically consists of several modules. The contribution barriers refer to a module’s ease of modifying and coding, variation of computer languages, and modularity. Krogh et al. (2003) found that a large number of developers specialized in “easy” rather than “hard” modules. It is conceivable that developers in “easy” modules can hardly communicate with and coordinate knowledge transferred from developers in “hard” modules because “easy” modules’ developers do not have proper capability to absorb the knowledge. Therefore, we hypothesize:

H2: Knowledge differentiation is negatively associated with knowledge coordination behaviors of the members of an OSS team.

Knowledge credibility

We conceptualize knowledge credibility as the third dimension of TMS and define knowledge credibility as the extent to which the developers of an OSS team have confidence in each other’s knowledge. 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 to coordinate effectively. This trust makes team members willingly to internalize knowledge from others (Joshi et al., 2004; Joshi et al., 2005), allows team members to carry out tasks of their specialties without explicitly justifying their course of action (Liang et al., 1995), and avoid criticizing each other’s work too often (Moreland, 1999). Therefore we hypothesize:

H3: Knowledge credibility is positively associated with knowledge coordination behaviors of the members of an OSS team.

OSS team performance

Previous research has shown that OSS team performance is attributable to a number of factors. For instance, Gallivan (2001) claimed that OSS team effectiveness might be dependent upon social and self-control mechanisms, such as individual reputation and membership management. Stewart and Gosain (2006) carried out a field study to examine the relation between the OSS ideology and OSS team effectiveness. The results showed that the OSS ideology affected team effectiveness through communication quality, affective trust, and cognitive trust.

Grewal et al. (2006) suggested 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) (Stewart and 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). We agree 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. Therefore, this study narrows its focus on the technical achievement of the team performance.

Several researchers have substantiated the positive relation between knowledge coordination and technical achievement in the software development team settings. For example, Faraj and Sproull (2000) found that expertise coordination improves the software development team’s work quality. Tiwana (2004) also reported that knowledge integration is positively associated with the reliability of software produced because teams with effective knowledge coordination incur fewer misunderstanding and confusion. Therefore, we hypothesize:

H4: Knowledge coordination within an OSS team positively affects the team’s technical achievement.

Figure 1 summarizes the hypotheses discussed above.
METHOD

This study mainly adopted a cross-sectional survey design as its method.

Sample

Data for this study was gathered from OSS teams hosting projects on Sourceforge.net (SF). Currently, SF has registered more than 100,000 projects. These projects are broadly classified into fourteen categories: clustering, database, desktop, development, enterprise, financial, games, hardware, multimedia, networking, security, system administration, storage, and VoIP. We sampled projects from two randomly selected categories: clustering and system administration. Projects included in the sample must meet two criteria. First, since the study is concerned with a team-level phenomenon, the projects must have at least two developers. Second, to make sure that the abandoned projects were excluded from the sample, the team must have been active in the past 60 days at the time of the sample selection. At the end, we obtained a sample of 149 projects.

Measures

A Web-based survey was designed to measure knowledge location (KL), knowledge differentiation (KD), knowledge credibility (KCR), and knowledge coordination (KCO). Following Stone’s recommendation (1978), the survey items were developed by largely adapting previously validated items. Specifically, Lewis (2003) had developed the items measuring knowledge differentiation and credibility. These items had been validated in a variety of teams (e.g., student project teams and cross-functional teams). Thus, we adapted these items to measure KD and KCR in this study. Faraj and Sproull (2000) had developed and validated two sets of items to measure knowledge sharing and location in the context of software development teams. We adapted these items to measure KCO and KL in this study. All the items from earlier studies were reworded to fit in the current research context. All items use a 7-point Liker scale anchored from “strongly disagree” to “strongly agree.” Table 1 summarizes the measurement items.
member knows who on the team she or he should ask for the answer
- KL6 – Our members have a hard time identifying the experts on the team
- KL7 – Our members have no idea what special knowledge and expertise other members on the team possess

- KD1 – Each team member has specialized knowledge of some aspect of our project
- KD2 – Different team members are responsible for different domains of expertise needed for our project
- KD3 – Each team member has knowledge about some aspect of our project that no other team member on the team has
- KD4 – The specialized knowledge of several different members is needed to complete our project
- KD5 – Members of our team specialize in different aspects of the project (removed)
- KD6 – Members on our team have project-relevant knowledge that overlaps each other (removed)
- KD7 – Members on our team are “generalists” (removed)

- KCR1 – The members on our team do not have doubts on project-relevant suggestions from other members (removed)
- KCR2 – The members on our team trust that the other members’ knowledge about the project is credible
- KCR3 – The members on our team are confident when applying the knowledge provided by other members to the project tasks at hand
- KCR4 – The members on our team did not have much faith in the other members’ “expertise”
- KCR5 – The members on our team like to double-check the knowledge provided by other members before applying it to the project tasks at hand (removed)

- KCO1 – Members in our team share their special knowledge and expertise with one another
- KCO2 – 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 (removed)
- KCO3 – Members in our team virtually do not share their information, knowledge, or skills with one another
- KCO4 – More knowledgeable members in our team willingly make their knowledge and expertise available to other members
- KCO5 – Project tasks are completed by integrating the specialized knowledge of different members in our team (removed)

Table 1. Measures

In addition, we objectively measured technical achievement using the percentage of closed project issues (i.e., bugs, feature requests, and patches) relative to the total project issues (Stewart and Gosain, 2006). Such information was publicly available from the sampled projects’ websites.

Data collection

Over a period of two weeks in October 2007, we sent out a series of three emails, enclosing the Web survey link, to invite project administrators from 149 projects to participate in the study. The administrators are typically either the initiators or major code contributors of the projects (Moon and Sproull, 2000). Thus, they should be “more familiar with the team’s internal dynamics, activities, and accomplishments” than other OSS team members (Stewart and Gosain, 2006, p. 299) and be in the best position to assess team-level perceptions. Of those invited, project administrators from 61 projects completed the survey, yielding a response rate of 40.94%.
After the survey administration phase, project data of the responding projects was manually collected from their websites. Specifically, we collected the number of project issues, both opened and closed, reported from the beginning of the project till the date when the survey was completed to measure technical achievement of the OSS team.

ANALYSIS AND RESULTS

Reliabilities and validities

Because the survey items were largely adapted from prior studies, their reliabilities and validities had to be reestablished in the current research context. Therefore, the survey items were first subjected to reliability assessment using Cronbach’s alpha. A scale with an alpha of 0.7 is considered adequately reliable (Cronbach, 1951). To achieve this alpha, the items were dropped from the following scales due to their poor item-scale correlations: knowledge location (KL2 and KL3), knowledge differentiation (KD5, KD6, and KD7), knowledge credibility (KCR1 and KCR5), and knowledge coordination (KCO2 and KCO5). Table 1 shows the reliabilities of all the scales after unreliable items were removed.

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). However, a more recent study (i.e., Stewart and Gosain, 2006) provides and demonstrates 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.

Using the above solution, the items of the current study were divided into two subsets. One subset included the items measuring knowledge location and knowledge credibility, and another included the items measuring knowledge differentiation and knowledge coordination. The principal component analysis with the varimax rotation was then conducted on each subset.

Factor analysis on items related to knowledge location and knowledge credibility yielded two factors (see Table 2). These two factors were consistent with the two constructs that these items were designed to tap. All items had very good loadings on their intended factors with minimum cross-loadings. Therefore, the convergent and discriminant validities of these items were established.

<table>
<thead>
<tr>
<th>Items</th>
<th>Factor 1</th>
<th>Factor 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>KL1</td>
<td>0.69</td>
<td>0.21</td>
</tr>
<tr>
<td>KL4</td>
<td>0.66</td>
<td>0.35</td>
</tr>
<tr>
<td>KL5</td>
<td>0.89</td>
<td>-0.07</td>
</tr>
<tr>
<td>KL6</td>
<td>0.74</td>
<td>0.35</td>
</tr>
<tr>
<td>KL7</td>
<td>0.79</td>
<td>0.26</td>
</tr>
<tr>
<td>KCR2</td>
<td>0.28</td>
<td>0.72</td>
</tr>
<tr>
<td>KCR3</td>
<td>0.02</td>
<td>0.92</td>
</tr>
<tr>
<td>KCR4</td>
<td>0.33</td>
<td>0.76</td>
</tr>
</tbody>
</table>

Table 2. Validity of KL and KCR Items

Factor analysis on items for knowledge differentiation and knowledge coordination yielded two factors (see Table 3). These two factors concurred with the two constructs that these items were designed to measure. All items had very good loadings on their intended factors with minimum cross-loadings. Therefore, the convergent and discriminant validities of the items were demonstrated.
<table>
<thead>
<tr>
<th>Factor</th>
<th>Item</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>KD1</td>
<td></td>
<td>-0.04</td>
<td>0.70</td>
</tr>
<tr>
<td>KD2</td>
<td></td>
<td>0.05</td>
<td>0.84</td>
</tr>
<tr>
<td>KD3</td>
<td></td>
<td>-0.17</td>
<td>0.73</td>
</tr>
<tr>
<td>KD4</td>
<td></td>
<td>0.34</td>
<td>0.72</td>
</tr>
<tr>
<td>KCO1</td>
<td></td>
<td>0.92</td>
<td>-0.17</td>
</tr>
<tr>
<td>KCO3</td>
<td></td>
<td>0.85</td>
<td>-0.01</td>
</tr>
<tr>
<td>KCO4</td>
<td></td>
<td>0.76</td>
<td>-0.14</td>
</tr>
</tbody>
</table>

Table 3. Validity of KD and KCO Items

Test of the Hypotheses

We used the partial least squares (PLS) approach to test the proposed hypotheses. According to the widely accepted rule of thumb (Chin, 1998), the PLS analysis for this study required minimally 30 observations. Since we collected 61 observations, the sample size was well beyond the minimum sample size recommended.

We assessed the proposed hypotheses by evaluating the coefficients of the corresponding paths shown in Figure 1. We used the bootstrapping method with the sample size of 61 and 500 resamples (Tenenhaus et al., 2005). The results are summarized in Table 4.

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Path Coefficients (Standard Error)</th>
<th>Results (t-value; p-value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1: Knowledge Location ( \rightarrow ) Knowledge Coordination</td>
<td>0.39* (0.14)</td>
<td>Yes (2.87; ( p &lt; 0.05 ))</td>
</tr>
<tr>
<td>H2: Knowledge Differentiation ( \rightarrow ) Knowledge Coordination</td>
<td>-0.20* (0.10)</td>
<td>Yes (-2.07; ( p &lt; 0.05 ))</td>
</tr>
<tr>
<td>H3: Knowledge Credibility ( \rightarrow ) Knowledge Coordination</td>
<td>0.08 (0.15)</td>
<td>No (0.50; ( p &gt; 0.10 ))</td>
</tr>
<tr>
<td>H4: Knowledge Coordination ( \rightarrow ) Technical Achievement</td>
<td>0.19† (0.11)</td>
<td>Marginally Yes (1.74; ( p &lt; 0.10 ))</td>
</tr>
</tbody>
</table>

Table 4. Hypotheses Testing

H1, H2, and H3, collectively, suggested the relation between TMS and knowledge coordination among the members of an OSS team. The PLS results show that the path coefficient from knowledge location to knowledge coordination (H1) is positive and significant (\( b = 0.39, t\text{-value} = 2.87, p < 0.05 \)), thus lending support for H1. The path coefficient from knowledge differentiation to knowledge coordination (H2) is negative and significant (\( b = -0.20, t\text{-value} = -2.07, p < 0.05 \)). Therefore, H2 is also supported. Finally, the path coefficient from knowledge credibility to knowledge coordination (H3) is not significant (\( b = 0.08, t\text{-value} = 0.50, p > 0.10 \)). Therefore, H3 is not supported.
H4 proposed a positive relation between knowledge coordination and technical achievement of an OSS team. The results report a positive and marginally significant coefficient for this path \((b = 0.19, t\text{-value} = 1.74, p < 0.10)\), thus marginally supporting H4.

**DISCUSSION**

To our knowledge, this is the first study to examine TMS in OSS teams. The study has validated the important role that TMS plays in OSS developers’ knowledge coordination behaviors, which, in turn, have positive influence on their projects’ technical achievement. The results indicate that knowing the location of knowledge distributed among the members of an OSS team helps the team coordinate its knowledge effectively. The importance of knowledge location is consistent with the finding of Kanawattanachai and Yoo (2007). In addition, the results show that allowing OSS developers to specialize in different knowledge domains has detrimental effects on their knowledge coordination behaviors. Therefore, it is necessary to have a few “generalists” in OSS project teams. Such roles are often assumed by project administrators. They have been working on projects for a long time and are familiar with the overall development of projects, and thus can help integrate specialized knowledge from different team members to bear on software development tasks.

Interestingly our results show that knowledge credibility has no impacts on knowledge coordination behaviors of OSS developers. This finding is inconsistent with early studies (Joshi et al., 2004, Joshi et al., 2005). They found that knowledge credibility is an important antecedent for knowledge transfer and coordination behaviors. However, these studies used student project teams as the subjects, which are different from OSS project teams in a significant way. That is, OSS project teams are comprised of voluntary developers. Previous OSS literature (e.g., Roberts et al., 2006) has identified that factors, such as altruism, pro-sharing norms, and reciprocity are the major reasons why the developers voluntarily join and contribute coding to the OSS community. We speculate that because of these motivational factors and volunteer nature of the work, OSS developers presume that everyone in the team acts on goodwill, and thus seek and accept knowledge from others without assessing its credibility.

Lastly, our results indicate that knowledge coordination has some positive bearing the technical achievement of an OSS team. In other words, more effective an OSS team coordinates its knowledge, the more bugs the team is able to resolve.

**LIMITATIONS AND FUTURE DIRECTIONS**

Despite the contribution to the literature, this study has several limitations. First, this study has demonstrated the importance of TMS in OSS project teams. However, we did not investigate how the TMS is developed at the first place. Some prior research (Kanawattanachai and Yoo, 2007; Moreland, 1999) suggested that factors, such as task-oriented communication and shared task experience, affect the development of TMS. But these factors have not been adequately tested in virtual teams, such as OSS project teams. Therefore, an important direction for future research is to include these factors in this study’s research model and then test the extended model in the context of OSS teams.

Second, our sample was exclusively drawn from two project categories on the SF website: clustering and system administration, which limits the generalizability of the study’s findings. Thus, another direction for future research is to replicate this study in other project categories.

**REFERENCES**


