doi:10.1111/j.1365-2575.2009.00347.x

The factors that affect the performance of open source software development – the perspective of social capital and expertise integration

Shih-Wei Chou* & Mong-Young He†

*Department of Information Management, National Kaohsiung First University of Science and Technology, 2 Juoyue Road, Nantz District, Kaohsiung 811, Taiwan, email: swchou@ccms.nkfust.edu.tw, and †Department of Management, National Kaohsiung First University of Science of Technology, 2 Juoyue Road, Nantz District, Kaohsiung 811, Taiwan, email: mvdis2@kcg.gov.tw

Abstract. *Drawing on social capital theory, we develop a theoretical model aiming to explore how open source software (OSS) project effectiveness (in terms of team size, team effort and team's level of completion) is affected by expertise integration. This in turn is influenced by three types of social capital – relational capital, cognitive capital and structural capital. In addition, this study also examines two moderating effects – the impact of technical complexity on the relationship between cognitive capital and expertise integration, and of task interdependence on the relationship between expertise integration and task completion. Through a field survey of 160 OSS members from five Taiwanese communities, there is support for some of the proposed hypotheses. Both reciprocity and centrality affect expertise integration as expected, but the influence of commitment and cognitive capital (including expertise and tenure) on expertise integration is not significant. Finally, expertise integration affects both team size and team effort, which in turn jointly influence task completion. This research contributes to advancing theoretical understanding of the effectiveness of free OSS development as well as providing OSS practitioners with insight into how to leverage social capital for improving the performance of OSS development.*

Keywords: open source software (OSS), information systems (IS) development, social capital theory

1. INTRODUCTION

There has been a surge of interest among academics and practitioners in open source software (OSS) development over the last decade, and one of the reasons for this is because

© 2010 Blackwell Publishing Ltd

OSS has helped companies achieve greater penetration of the market and offered an opportunity for firms to establish an industry standard, thus increasing competitive advantage (Sharma *et al.*, 2002; Lee *et al.*, 2009). While some of the OSS development projects (e.g. Mozilla Web browser, OpenOffice productivity suite) are performed by loosely coordinated software developers such as free OSS, others such as MySQL and Linux operating systems are tightly controlled by for-profit organizations and have full-time paid development teams. This study focuses on the free OSS projects because most OSS developers work on projects that do not typically have a corporate owner to organize and improve the development processes being put into practice (Scacchi *et al.*, 2006). Empirical evidence is beginning to emerge that establishes the viability and effectiveness of the OSS development paradigm (Wu *et al.*, 2007). They predicted that OSS products will be well established by 2010 in at least 75% of mainstream enterprises.

To facilitate OSS development, studies suggest leveraging and organizing social relationships in open source communities so that high-quality software can be produced in a relatively short period of time, with very little cost, by some of the best programmers in the profession (Bergquist & Ljungberg, 2001; Sharma *et al.*, 2002; Napier *et al.*, 2009). Free software and OSS are often treated as the same thing. However, there are important differences between them regarding the beliefs/ideologies of their practitioners as to how and why software should be developed for sharing and modification (Scacchi *et al.*, 2006). Free software is a social movement, while OSS development is a software development methodology based on Free Software Foundation. As noted previously, this study focuses on free OSS development because most OSS teams are composed of volunteers working without financial remuneration directly tied to their output.

One of the most common questions about free OSS development projects is why software developers will join and participate in such efforts, often without pay for sustained periods of time (Stewart & Gosain, 2006; von Krogh & Spaeth, 2007). According to Scacchi *et al.* (2006), the reasons for participating in free OSS development are twofold. First, participants have a greater opportunity for learning and for sharing what they know about information systems (IS) functionality, design and practices associated with specific projects. Second, since free OSS developers self-select the technical roles they will take on as part of their participation in a project, rather than being assigned to a role in a traditionally managed IS project, where the assigned role may not be to their liking, they are more likely to enjoy their OSS work and to be recognized as trustworthy and reputable contributors.

Research suggests that understanding the features of OSS development in OSS communities may help identify the key factors that affect the performance of OSS development (Stewart and Gosain, 2006; von Krogh & Spaeth, 2007). Sharma *et al.* (2002) propose an OSS framework aimed at comprehending the process of creating and sustaining OSS communities based on organization theory. According to Sharma *et al.*, OSS communities use coordination mechanisms that emphasize decentralized workspaces and asynchronous communication because OSS developers are geographically distributed and cannot devote large blocks of time to the project in a consistent manner (Markus *et al.*, 2000). Furthermore, because there is often no monetary compensation to be expected for efforts conducted in free OSS development (Bergquist & Ljungberg, 2001), the will to contribute to the community has to be explained in terms of other salient factors such as gift cultures rather than being based on traditional cost–benefit rationality. Gift cultures are based on gift economics, in which social relationships are not regulated by the possession or exchange of commodities. Instead, gift cultures are characterized by the creation and maintenance of social relationships based on the economy of gift exchange (Scacchi *et al.*, 2006).

Based on the above analysis, social networking, interconnecting multiple OSS projects and coordinating OSS developers' efforts all play an important role in OSS success (Bergquist & Ljungberg, 2001; Sharma *et al.*, 2002; von Krogh & Spaeth, 2007). Although the above antecedents have been identified by prior work (Tiwana & McLean, 2005; Scacchi *et al.*, 2006), a systematic approach to this remains absent. To fill this gap, this study examines free OSS development from an alternative perspective based on social capital theory, which is appropriate for explaining gift economics and sharing practices between software developers, and paradoxical phenomenon of collective action as is the case with free OSS development (Bergquist & Ljungberg, 2001; Stewart & Gosain, 2006). For example, contributions of knowledge to electronic networks seem irrational because giving away knowledge eventually causes the possessor to lose his or her unique value relative to what others know, and benefits all others except the contributor (Wasko & Faraj, 2005). This is similar to the context of free OSS development, in which developers are involved in software development in the absence of traditional organizational incentives and controls, and they voluntarily contribute their time, effort and expertise towards the collective benefit, when they can easily free ride on the efforts of others. In addition to social capital, as a special type of IS development (ISD), the effectiveness of OSS development depends on how members of an OSS team coordinate and integrate their specialized expertise or social capital to jointly develop project concepts, designs and solutions (Tiwana & McLean, 2005). This was conceptualized as the key factor that mediates the relationship between social capital and the success of OSS (Stewart & Gosain, 2006). Finally, theory also suggests that the success of IS implementation is contingent on key antecedents such as technical complexity and task interdependence (Karimi *et al.*, 2004; Sharma & Yetton, 2007). In sum, we seek to answer three research questions: (1) What are the factors, in terms of social capital, determining OSS team effectiveness? (2) How does expertise integration mediate the relationships between social capital and OSS team effectiveness? (3) What are the influence of other salient factors of ISD such as technical complexity and task interdependence?

2. THEORY AND HYPOTHESIS DEVELOPMENT

2.1 Effectiveness in OSS development teams

Figure 1 lists the research model of this study. As with the study of commercial software development contexts, a multidimensional view on team effectiveness is important for a free OSS project (Crowston *et al.*, 2003). Following Stewart & Gosain's (2006) study, we focus on

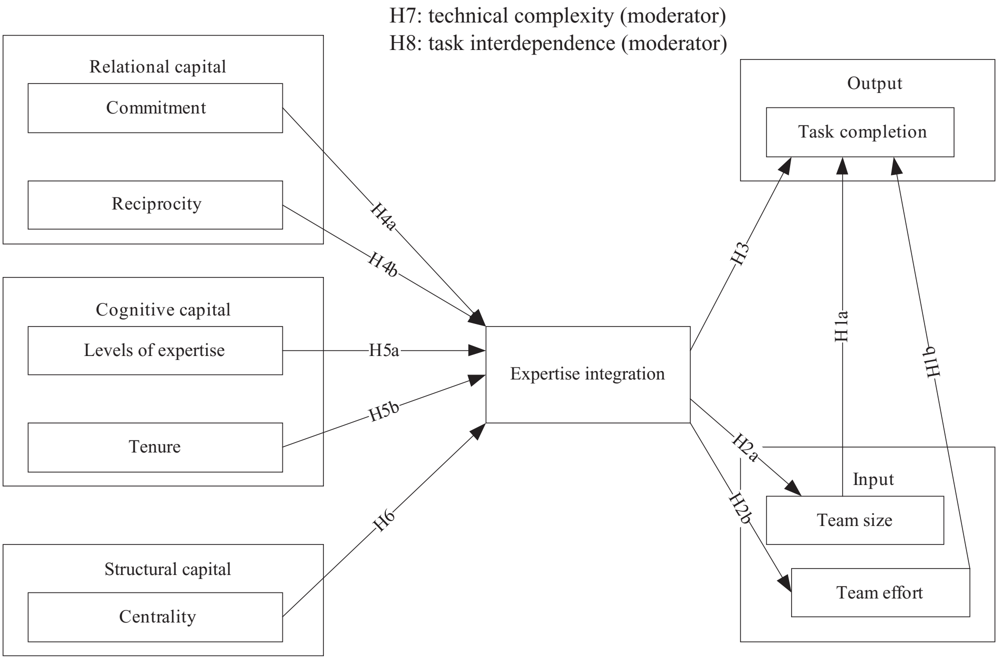

Figure 1. Research model.

two aspects of OSS effectiveness – input to an OSS community and output of an OSS project. OSS input refers to the extent to which an OSS project attracts input to the development community such as the number of developers associated with the project (i.e. team size) and the total number of workweeks devoted to the project (i.e. team effort). OSS output refers to the observable output produced by an OSS project such as the addition of new features to the software or the fixing of software bugs. While commercial projects have employees paid and directed through formalized mechanisms, one of the main concerns identified is how to attract developers and motivate their input to the OSS project (von Krogh & Spaeth, 2007). Since the success of free OSS relies on members donating their efforts voluntarily, the amount of input to an OSS project (i.e. how many people devote how much effort) is an important aspect of effectiveness. Thus, an OSS team's input effectiveness includes two parts – the number of developers that have been attracted and retained to work on the team (team size) and an estimate of the amount of effort those OSS members have devoted to the team (team effort).

Given that budget constraints are not the major concern for free OSS projects (Scacchi, 2002), a more appropriate measure of OSS outcome may be that related to the ongoing productivity of the team. ISD teams rely on change management systems to organize and track their development work, with actual changes to code being associated with a basic unit of work called a modification request. Empirical studies report that analyses of ISD in distributed contexts and of OSS projects such as Apache and Mozilla (Mockus *et al.*, 2002; Herbsleb &

Mockus, 2003) have used responses to modification requests as indicators of work accomplishment. Following their work, this study uses task completion, which refers to the extent to which the team completes identified work tasks, as the output effectiveness construct.

Input-process-output models of teamwork (Ilgen *et al.*, 2005) suggest that the effectiveness of a free OSS development team in attracting and retaining input tends to affect its output effectiveness. Given members of an OSS project are likely to specialize in working on small parts of an overall project (von Hipple & von Krogh, 2003), having a larger pool of developers indicates a larger pool of available specializations and a greater chance that the expertise needed for completing a variety of tasks will be available. This leads to Hypothesis 1a:

H1a: An OSS project's team size will have a positive effect on the team's level of task completion.

Similarly, the more effort that OSS members devote to the project, the greater the number of tasks they will be able to complete (Stewart & Gosain, 2006). This leads to Hypothesis 1b:

H1b: An OSS project's team effort will have a positive effect on the team's level of task completion.

2.2 Expertise integration and the effectiveness of OSS projects

A number of factors may affect the effectiveness of OSS projects, including OSS developers' motivations and intentions to continue their involvement, the structure of OSS communities (division of labour, coordination mechanisms and distribution of decision-making authority), governance processes, development processes, the culture of OSS communities and ideology (beliefs, values and norms) (Bergquist & Ljungberg, 2001; Sharma *et al.*, 2002; Stewart & Gosain, 2006; Wu *et al.*, 2007). This study focuses on the relationship between the integration and coordination of individual OSS developers' expertise because this plays an important role in both traditional IS and free OSS development, and the impact of expertise integration on OSS development is different from that on traditional ISD.

The major processes within OSS communities can be classified into governance and software development (Sharma *et al.*, 2002). Regarding the self-governance of OSS communities, the initial software developer maintains a lead role, but formal authority is vested in a team. Projects are partitioned by lead architects (or designers) into manageable modules and handled by individuals and teams. Lead architects are responsible for the coordination and synthesis of teams. In addition to the self-governance mechanisms, the development processes in an OSS project are different from those of traditional IS. OSS developers iterate through a common series of actions while working on the software source, including problem discovery, finding volunteers, solution identification, code development and testing, code commitment and documentation, and release management. According to Scacchi *et al.* (2006), the initial OSS software developer serves as a central node in a social network that interconnects multiple free OSS projects, which is similar to accumulating social capital and recognition from peers. The success of such OSS development depends on how well the merger of independent free OSS systems into larger composite ones performs (Madey *et al.*, 2004). In other words, gaining the critical mass of core developers to grow more substantially and attracting user–developer communities play a key role in the success of free OSS development.

To explain how to facilitate the free OSS development, we use a knowledge-based view of organizing (Grant, 1996). Based on this view, OSS project teams are used as vehicles for integrating knowledge that is distributed among many OSS contributors (Sharma *et al.*, 2002; Scacchi *et al.*, 2006). In order for an OSS project to benefit from individuals' expertise, it is necessary to integrate specialized, individually held expertise into collective (project) knowledge (Okhuysen & Eisenhardt, 2002). Following prior studies (Grant, 1996; Tiwana & McLean, 2005), we define expertise integration as the coordinated application of individually held expertise in the accomplishment of tasks at the project level.

Fully understanding the problem that the intended system must solve is often one of the most challenging aspects of ISD (Tiwana & McLean, 2005); this is also the case with OSS development (Stewart & Gosain, 2006). Two types of knowledge are necessary when addressing the problem domain – technical knowledge and knowledge about the application problem domain. Given knowledge is dispersed among different project stakeholders of an OSS project, including owner (analysts, domain experts), developers (programmers) and potential users. Unlike traditional software development, users and developers are often one and the same in open source. Thus, how OSS developers' technical and application domain knowledge can be effectively integrated at the project level in formulating project concepts and solutions becomes an important issue (Bergquist & Ljungberg, 2001; Hickey & Davis, 2004). Even when some of the above knowledge is available within the team, it is not readily available in an explicit form such as requirements documents or formal specifications. Besides, OSS development largely consists of heuristic tasks, i.e. tasks that do not have clear and readily identifiable paths to the solutions (Cooper, 2000; Herbsleb & Mockus, 2003; Scacchi *et al.*, 2006). Thus, unless both technical and application domain knowledge are integrated during the OSS development process, it is less likely that artefacts of the OSS development process (e.g. specifications, features and code) will capture the capabilities that users need from the system (Okhuysen & Eisenhardt, 2002).

Since expertise integration positively affects the success of OSS, more integration of team members' expertise at the project level implies that the OSS project is more likely to be successful, which in turn attracts and retains more developers (i.e. larger team size). This leads to Hypothesis 2a:.

H2a: An OSS team's expertise integration will have a positive effect on team size.

Similar to H2a, when more expertise integration is accomplished by an OSS team, developers are more willing to devote their time and effort to the OSS project because they believe expertise integration is more likely to result in a successful project. This leads to Hypothesis 2b:

H2b: An OSS team's expertise integration will have a positive effect on team effort.

Finally, as noted previously, task completion of an OSS project relies on how effectively it achieves responses to modification request in distributed ISD contexts (Herbsleb & Mockus, 2003; Stewart and Gosain, 2006). Because expertise integration is likely to lead to formulation of project concepts and solutions by integrating developers' technical and domain knowledge and addressing heuristic tasks (Tiwana & McLean, 2005), it may positively affect an OSS team's capability of handling responses to modification requests or task completion. This leads to Hypothesis 3:

H3: An OSS team's expertise integration will have a positive effect on the team's level of task completion.

2.3 Social capital and expertise integration

Theory suggests that collective action is likely to be affected by social capital (Coleman, 1990), which refers to 'resources embedded in a social structure that are accessed and/or mobilized in purposeful action' (Lin, 2001). Empirical findings support this argument (Wasko & Faraj, 2005). Their study shows that knowledge contribution is affected by three different types of social capital – structural capital, cognitive capital and relational capital. Tiwana & McLean (2005) found that relational capital, in terms of trust and reciprocity, affect expertise integration. In the context of OSS, expertise integration not only entails contributing explicit knowledge but also ensures that the tacit elements of knowledge can be integrated and utilized at the project level in formulating project concepts and solutions. Expertise integration represents a type of collective action, in which effective teamwork emerges from new knowledge that results from interactions among specialists in a team, not simply from individual gains in knowledge by individual team members (Okhuysen & Eisenhardt, 2002). Thus, the viability of expertise integration is likely to be affected by social capital. This study extends prior work (Tiwana and McLean, 2005; Wasko & Faraj, 2005) by examining three types of social capital in the context of OSS development – relational, cognitive and structural.

2.3.1 Relational capital

Relational capital refers to the affective nature of the relationships within a group (or community) (Nahapiet & Ghoshal, 1998), including strong identification with the group, trusting others within the collective, feeling obliged to participate in the collective, and recognizing and abiding by its cooperative norms (Coleman, 1990; Wasko & Faraj, 2005). According to Wasko & Faraj (2005), the main purpose of relational capital is to facilitate actions for individuals within the groups, and relational capital is an important asset that benefits both the community and its members. Members are willing to help other members, even strangers, simply because they are part of the collective and all have a collective goal. Although relational capital involves several key factors, following prior work (Tiwana & McLean, 2005; Wasko & Faraj, 2005), this study examines two aspects of relational capital – commitment and reciprocity, because they are more likely to affect the expertise integration of an OSS team.

Commitment represents a duty or obligation to participate in future action and arise from frequent interaction (Coleman, 1990). Although commitment is often used for describing direct expectations developed within particular personal relationships, it can also accrue to a collective. Commitment to a collective, such as an OSS community, conveys a sense of responsibility to help others within the collective on the basis of shared membership. As suggested by Constant *et al.* (1996), in an organizational electronic network, individuals post valuable advice because of a sense of obligation to the organization. Similarly, individuals participating in extra-organizational electronic networks are motivated by a perceived moral obligation to pay back the network and the profession as a whole (Wasko & Faraj, 2005). Thus, OSS members who feel a strong sense of commitment to the community are more likely to consider it a duty to assist other members and contribute knowledge. Besides, commitment also strengthens the ties among members, which in turn reduce the costs of sharing and integrating complex tacit knowledge. Finally, this sense of commitment increases members' willingness to build on each other's perspectives, ideas and expertise during the processes of OSS development because they feel they are responsible for the success of OSS development. Thus, this leads to Hypothesis 4a.

H4a: Members' commitment to the OSS community will have a positive effect on expertise integration of OSS teams.

A basic norm of reciprocity is a sense of mutual indebtedness, so that individuals tend to reciprocate the benefits they receive from others, ensuring ongoing exchanges of views or ideas that benefit the individuals who are involved in the exchanges (Shumaker & Brownell, 1984). Even though exchanges in free OSS communities or other types of electronic networks of practice occur through weak ties between strangers, there is evidence of reciprocal supportiveness (Bergquist & Ljungberg, 2001; Wasko & Faraj, 2005; Scacchi *et al.*, 2006). Researchers (Sharma *et al.*, 2002; Tiwana & McLean, 2005) suggest that precise expertise contributions of each individual can be difficult to predict *ex ante*, although the general nature of each individual's contributions is predictable based on his or her assigned role in the OSS project. The norm of reciprocity facilitates contributions of expertise beyond levels that can be negotiated in advance. When there is a strong norm of reciprocity in a group, members believe that their knowledge contribution efforts will be reciprocated, thereby facilitating project-level integration of diverse ideas, perspectives and expertise that individual team members bring to the project. This leads to Hypothesis 4b:

H4b: Members guided by a norm of reciprocity will have a positive effect on expertise integration of OSS teams.

2.3.2 Cognitive capital

Cognitive capital refers to those resources that make possible shared interpretations and meaning within a collective (Nahapiet & Ghoshal, 1998). Following prior work (Wasko & Faraj, 2005), cognitive capital consists of both individual expertise and experience with applying the expertise. To facilitate expertise integration, it is necessary for members to have at least some level of shared understanding between them, such as shared language and vocabulary.

Language is the means by which individuals participate in knowledge exchange. It provides a frame of reference for interpreting the environment, and its mastery is typically indicated by an individual's level of expertise (Wasko & Faraj, 2005). Expertise integration of OSS development is likely to be facilitated by an individual's expertise or mastery of the language within the practice because the more project-related expertise (technical and domain knowledge) an individual has, the more likely he or she can understand the context in which his or her knowledge is relevant (Brown & Duguid, 1991; Scacchi *et al.*, 2006). Furthermore, individuals with higher levels of expertise are more likely to provide more useful knowledge and advice on how to build on other's perspectives and ideas, leading to more project-level integration of diverse ideas. Thus, individuals' expertise in the shared practice may affect the integration of expertise that individual team members bring to the project. This leads to Hypothesis 5a:

H5a: Members' levels of expertise in the shared practice will have a positive effect on expertise integration of OSS teams.

As noted previously, cognitive capital also consists of mastering the application of expertise, which takes experience (Coleman, 1990). Individuals with longer tenure in the shared practice are likely to better understand how their expertise is relevant to others' ideas and expertise during the OSS processes (Sharma *et al.*, 2002; Tiwana & McLean, 2005). This indicates that they are better able to share knowledge with others and achieve integration of diverse ideas and perspectives that individual team members bring to the project. Thus, this leads to Hypothesis 5b:

H5b: Members' tenure in the shared practice will have a positive effect on expertise integration of OSS teams.

2.3.3 Structural capital

Theories of collective action and social capital (Coleman, 1990; Nahapiet & Ghoshal, 1998) suggest that the connections between individuals, or the structural links created through the social interactions between individuals in a network, play a key role in predicting collective action. When networks are dense, consisting of a large portion of strong, direct ties between members, collective action, such as finding novel associations and linkages among the diverse ideas and domain expertise that individual team members hold, is relatively easy to achieve (Tiwana & McLean, 2005; Wasko & Faraj, 2005). The more individuals are in regular contact with one another, the more likely they are to develop a 'habit of cooperation' and act collectively (Marwell & Oliver, 1993).

Structural capital is also relevant to individual actions, such as contributing to individuals' technical and domain knowledge of an OSS project (Sharma *et al.*, 2002; Scacchi *et al.*, 2006). Individuals who are centrally embedded in a collective imply that they have a relatively high proportion of direct ties to other members. Thus, they are more likely to have developed this habit of cooperation, indicating they are more willing to comply with group norms and expectations – i.e. social ties (Ahuja *et al.*, 2003; Wasko & Faraj, 2005). From an OSS development viewpoint, this implies that more ideas, perspectives and expertise can be brought to the project – i.e. members have more opportunity to access a variety of alternatives, examples, solutions and ideas. Habit of cooperation also indicates that individuals are more likely to better understand what ideas are relevant to the project and how individually held expertise can be applied to project activities with an appreciation of the project context, its business needs and its constraints (Cooper, 2000; Tiwana & McLean, 2005). Thus, members' network centrality (social ties) may positively affect expertise integration.

Following prior work (Ahuja *et al.*, 2003; Wasko & Faraj, 2005), this study conceptualizes structural capital as the number of social ties the individual has with others in the network. Social ties refer to social interaction with other members of a network, which entails conversations between members and exchanges of views. Because a social tie or structural link is created when a member responds to another's message (or posting messages), the degree of her centrality depends on how many social ties she creates. Consequently, we expect that expertise integration is positively associated with members' centrality. This leads to Hypothesis 6:

H6: Members with higher levels of network centrality will have a positive effect on expertise integration of OSS teams.

2.4 The moderating role of technical complexity and task interdependence

2.4.1 Technical complexity

Technical complexity has been viewed as a key factor that affects the cooperation and effective knowledge transfer (Rogers, 1983; Sharma & Yetton, 2007). Following Sharma & Yetton (2007), this study defines technical complexity as those OSS implementation tasks that require advanced skills or need expert support to successfully accomplish. Knowledge embedded in individual cognitions plays a key role in task performance. Technically, complex IS implementation usually requires IS developers to work with unfamiliar technologies and to perform their tasks in different ways (Robey *et al.*, 2002). This also requires enhancement to the content of individual cognitions to overcome increased knowledge barriers (Fichman & Kemerer, 1997). The more technical complexity an IS team is confronted with, the more likely members of the team have more possible shared interpretations and meanings, which implies integration of the project-related ideas and the appreciation of project context and its business needs become more difficult (Cooper, 2000). Thus, technical complexity may affect the relationship between cognitive capital (expertise and tenure) and expertise integration.

Before explaining how technical complexity affects expertise integration, it is necessary to understand the related roles of OSS actors in OSS development (Scacchi *et al.*, 2006). Five different types of actors have been identified. First, the owner of an open source project is the person (or group) who started the project and has the exclusive right, recognized by the community at large, to redistribute modified versions of the software. As the owner attracts contributors, i.e. people that discover the software and want to contribute to its development, he or she becomes more of a coordinator or project leader (Bergquist and Ljungberg, 2001). Second, a group of code developers write most of the code concerning new functionality, review submitted code and make most of the decisions about releases (Mockus *et al.*, 2002). Empirical data (von Krogh & Spaeth, 2007) shows that almost all the new functionality is implemented and maintained by a small group of core developers. Third, defect repair involves a much wider development community, an order of magnitude larger than the core group. The fourth type of OSS actors are product users who voluntarily provide answers of the questions proposed by other users. In addition to the above users, there are a huge number of OSS users who are not actively contributing to OSS development. However, creating a critical mass of users is important both for the usability of a system and for the construction of a symbolic attraction surrounding OSS development. Finally, commercial businesses tied to OSS projects provide additional resources for developing free components of the OSS, but even more importantly, it helps to promote the OSS packages and drive them into the mainstream. For example, companies make money out of OSS by distributing the OSS and bundling it with their own products. Among the above actors, this study focuses on core developers and how technical complexity and task interdependence affect their expertise integration, because they account for most of the total code base as is the case with Apache and Orbiten (Bergquist and Ljungberg, 2001).

As noted by Sharma *et al.* (2002), the development processes of OSS core developers include problem discovery, solution identification, code development and testing, and code change review. When technical complexity is high, the required knowledge embedded in individual cognitions and skills in performing the above tasks are much more complex, and the effort required for integrating individually held expertise may be salient to OSS project managers because they face high knowledge barriers when synthesizing insights (or expertise) from the multiple thought worlds of the team members – both producers and users (Tiwana & McLean, 2005; Scacchi *et al.*, 2006). Conversely, when technical complexity is low, it is relatively easier to transfer OSS members' knowledge about OSS development processes (such as identifying the solution for a specific problem) or expertise in the problem domain because the barriers to the understanding of the application and business context knowledge are low (Sharma *et al.*, 2002; Sharma & Yetton, 2007). This leads to Hypothesis 7a:

H7a: The higher the technical complexity, the weaker the relationship between members' levels of expertise in the shared practice and expertise integration.

Similarly, when technical complexity is high, a team manager may find the effort required for expertise integration to be salient because longer experience with applying expertise may lead to different interpretation of and diverse ideas for an OSS task. Conversely, when technical complexity is low, members' experience may not be salient to expertise integration as tasks are more likely to have clear and readily identifiable paths to the solutions (Tiwana & McLean, 2005; Sharma & Yetton, 2007). This leads to Hypothesis 7b:

H7b: The higher the technical complexity, the weaker the relationship between members' tenure in the shared practice and expertise integration.

2.4.2 Task interdependence

In addition to the moderating effect of technical complexity, this study also investigates whether OSS code developers' expertise integration is contingent on task interdependence. Based on Sharma *et al.*'s (2002) OSS model, OSS communities have established processes for decision-making. Decision rights are primarily vested in individuals, and most decisions are reached by consensus (Mockus *et al.*, 2002). To facilitate effective and efficient communication, a variety of mechanisms are available, including email and chat rooms that support asynchronous communication. OSS development involves socially complex and communication-intensive tasks such as requirements elicitation and project coordination (Stewart & Gosain, 2006).

According to Robey *et al.* (2002), the implementation of interdependent-use IS innovations is frequently accompanied by the introduction of new business processes that disrupt existing task routines. The level of difficulty in successfully implementing IS increases as the level of task interdependence increases (Hackathorn & Keen, 1981) because of the misfit between the routines embedded in existing inter-individual cognitions (Gallivan *et al.*, 2005). Applying this to the context of OSS development, task interdependence is more likely to incur task overlaps and duplication of effort, or difficulty with code integration (Stewart & Gosain, 2006). To implement OSS effectively, it is necessary for OSS developers to not only enhance the content of individual cognitions but also develop new routines to cope with new task interdependencies (Edmondson *et al.*, 2001; Sharma *et al.*, 2002) because fully understanding the problem domain entails knowledge about the interdependent relationships among all OSS developers' application problem domain.

When task interdependence is high, integration of individually held expertise at the OSS team level is more difficult because it entails team managers' knowledge about the interdependent relationships among all team members' work procedures (Bergquist and Ljungberg, 2001; Stewart & Gosain, 2006; Sharma & Yetton, 2007). Furthermore, higher task interdependence also entails more shared knowledge about the skills and application problem domain, which in turn result in more coordination of individual expertise. As a result, the effect of expertise integration on task completion is contingent on task interdependence. In contrast, when task interdependence is low, collaborative task knowledge is not critical to the understanding of the problem domain and team members can use artefacts of the OSS (e.g. documentation) independently (Tiwana & McLean, 2005; Scacchi *et al.*, 2006; Stewart & Gosain, 2006). Thus, the influence of expertise integration on task completion diminishes. Based on the above argument, we posit that the effect of expertise integration on task completion is adversely contingent on task interdependence. This leads to Hypothesis 8:

H8: The higher the task interdependence, the weaker the relationship between expertise integration and task completion.

3. RESEARCH METHODOLOGY

This study used survey method for collecting data and testing the proposed research model. We examined our hypotheses by applying partial least squares (PLS) method to the collected data.

In an attempt to include those respondents who were most likely to have implemented an OSS project, the sample was drawn from five different OSS communities, including 'Software Liberty Association of Taiwan', 'Taiwan personal home page union', 'Drupal Taiwan', 'Taiwan personal home page association', and 'phpbb creating community'. These communities aim to enrich the open source community by providing a centralized infrastructure for developers to control and manage OSS development. This study's unit of analysis is a team working on a specific OSS project, with a variety of team sizes ranging from four to 10 members. Several approaches have been suggested to measure team-level constructs – assessing individual perceptions as a representation of team beliefs and team discussion to arrive at a common assessment (Stewart & Gosain, 2006). Given the practical difficulty of accessing all members of an OSS team, we used a key informant approach.

Although the use of project administrators as key informants could potentially skew the findings, using key informants to collect data about larger social entities is a common practice in organizational research, either at the firm level or at the team level (Sparrowe *et al.*, 2001). The use of key informants requires a deliberate strategy to access respondents that possess special qualifications relevant to the research such as status or specialized knowledge. Research on the effectiveness of ISD usually elicits project-level data from project managers (Ethiraj *et al.*, 2005). In the context of free OSS development, project administrators represent the role that is best suited to providing details of an OSS project. Thus, project administrators (or leaders) were chosen as the key informants because they are the individuals who are most familiar with OSS teams' internal dynamics, activities and accomplishments (Stewart & Gosain, 2006).

3.1 Operationalization of constructs

We developed the items in the questionnaire by adapting measures that had been validated by prior studies. Specifically, the items for the three salient antecedents of social capital – relational capital, cognitive capital and structural capital – were developed based on relevant theories (Blau, 1964; Nahapiet & Ghoshal, 1998) and empirical studies (Tiwana & McLean, 2005; Wasko & Faraj, 2005). Relational capital includes both commitment and reciprocity – the measures of the latter were adapted from Constant *et al.* (1996), and commitment measures were adapted from Mowday *et al.* (1979). Structural capital was assessed by determining members' degree of centrality to the network, which is defined as the number of interaction(s) between a focal individual and other members of an OSS community, regardless of the total number of messages posted. For the sake of simplicity, centrality was assessed by self-rated connections with other members, rather than using a square social network matrix as did Wasko and Faraj's study (2005). Finally, cognitive capital was assessed by self-rated expertise and tenure in the field.

The items measuring the effectiveness of an OSS team, including input to a team and task completion of an OSS project, were adapted from Stewart & Gosain's (2006) research on the effectiveness in OSS development teams. Input to a team includes team size and team effort – the former was measured as the number of developers associated with the project, and the latter was measured in terms of the total number of work weeks that a leader has devoted to an OSS project. The selected OSS sites track a variety of OSS activities, including the number of requests for bug fixes, patches, support and new features on each OSS project. In addition, these sites also calculate the number of the foregoing requests that have an uncompleted (or open) status. Task completion refers to the percentage of tasks completed – (total requests − requests open) / total requests × 100%, or zero for projects without task request. This operationalization is in line with studies of team-based ISD that use change requests as a measure of IS work accomplishment (Herbsleb & Mockus, 2003). Following Tiwana & McLean (2005), we measured expertise integration by assessing the extent to which members of an OSS team synthesized their individual expertise at the project level, synthesized members' tacit knowledge and expertise in the problem domain of an OSS project, understood the OSS project from a systematic perspective and blended new project-related knowledge with what members already know (Grant, 1996). Finally, following prior work (i.e. Pearce *et al.*, 1992; Choi & Kim, 2002; Sharma & Yetton, 2007), technical complexity was measured by three questions, and the level of task interdependence was measured in terms of a six-item scale of the fit between the existing inter-individual cognitions and those required for effective performance of new technologies.

An English version of the questionnaire was first compiled and modified to suit the context of OSS implementation and then translated into Chinese by a bilingual research associate. The Chinese version of the questionnaire was verified and refined for its accuracy of translation by one Management of Information Systems professor and one doctoral student who not only are familiar with OSS development but also have carried out extensive research into the causes of OSS's effectiveness. We then conducted a pretest for face and content validity with three OSS developers who have extensive experiences of undertaking OSS projects. Next, the internal consistency and discriminant validity of the instrument were assessed. Cronbach's alpha values ranged from 0.73 to 0.89. Due to low item-to-total correlation (less than 0.50), one item from reciprocity and two items from task interdependence were dropped.

3.2 Data collection and analysis

The selected OSS projects for the study fell into two categories – communications [Bulletin Board Systems, chat, and "I seek you" (ICQ)] and multimedia (audio, video and graphics three-dimensional rendering) – because we aimed for the control of project's differences in different product categories. After selecting categories, we ensured that the OSS projects had involved contributing OSS development activities to the code repository in the past few weeks, including requests for bug fixes, support or page views (von Krogh & Spaeth, 2007). This was to ensure that the respondents were involved in ongoing OSS projects. Finally, the OSS teams with at least four developers had been selected for the study because we emphasize team processes and dynamics and such number of members is appropriate to this context as suggested by OSS-related research (Stewart & Gosain, 2006). A total of 300 projects satisfied all criteria.

We offered participants a chance for wining a $150 lottery and an opportunity to be informed of the results. Because many members of the OSS community are involved in more than one OSS project, the survey instructed them to respond to reference to the OSS project with which they were most involved. This led to six respondents whose projects did not fall into the communication or multimedia categories. These six projects emphasized software that aims to support other functional applications – they can be classified as utility applications. In all, 140 project leaders responded to the survey. Because the results of analyses of variance (ANOVAs) indicated that there is no significant difference in outcome measures between first-round and second-round respondents, the two sets of measures were merged for analysis, leading to a sample size of 160 (an overall response rate of 53.3%).

PLS was used because it refers to a structural equation modeling technique that simultaneously assesses the reliability and validity of the measures of a theoretical model's constructs and estimates the relationships among these constructs (Chin *et al.*, 1998). Although the measurement and structural parameters are estimated together, a PLS model is assessed and interpreted in two stages – examining the reliability and validity of the measurement model and examining the structural relationships.

4. RESULTS

4.1 Measurement model

Table 1 lists the respondents' demographic information. As the analysis was based on data collected from two categories of OSS products – 66 communications and 94 multimedia – we

Table 1. Demographic information of respondents (n = 114)

Measure	Item	Frequency	Percentage
Gender	Male	107	66.9
	Female	53	33.1
Age	≦20	19	11.9
	21–30	75	46.9
	31–40	59	36.9
	41–50	6	3.8
	>50	1	0.6
Education	High school	23	14.4
	University (4 years)	109	68.1
	Graduate school	28	17.5
Market segments (or domains) of OSS products	Communication	61	38.1
	Multimedia	55	34.4
	Utility application (support of functional applications)	44	27.5

OSS, open source software.

conducted ANOVAs to investigate whether the effectiveness of OSS teams was significantly different in variables of interest in the foregoing categories. The results of the ANOVAs did not indicate a significant difference in mean values of any of the dependent variables, therefore the two sets of OSS projects were pooled for further analysis – $p = 0.62$ for team size, $p = 0.55$ for team effort and $p = 0.77$ for task completion.

To validate our measurement model, three types of validity were assessed – content validity, convergent validity and discriminant validity. Content validity is established by ensuring that the measurement items are consistent with the extant literature. This was performed by interviewing leaders of OSS projects and pilot-testing the instrument as noted earlier. Convergent validity is verified based on the assessment of composite reliability and average variance extracted (AVE) from the measures (Hair *et al.*, 1998). Although a few studies employing PLS have used 0.5 as the threshold reliability of the measures, 0.7 is a recommended value for a reliable construct (Chin *et al.*, 1998). The measure of AVE that is greater than 0.5 indicates acceptability (Fornell & Larcker, 1981). From Table 2, composite reliability ranges from 0.845 to 1, and all of the AVEs are higher than 0.5. Thus, the reliability of our data is acceptable. The weights and loadings of the measures in our model are significant on their path loadings at the level of 0.01. Then the discriminant validity was validated by assessing the square root of the AVE as suggested by Fornell and Larcker (1981). The results of Table 3 confirm the discriminant validity because the square root of AVE for each construct is greater than the level of correlations involving the construct. The results of inter-construct correlations also show that each construct shares greater variance with its own measures than with other measures.

In addition to validity assessment, we also considered the common method variance (CMV), which refers to a potential threat to internal validity, especially to research using surveys that collect responses in a single setting. To deal with CMV, we used the following approaches. First, we collected data in two separate stages – with dependent and independent variables measurement separated in time. Second, we used factor analysis to examine the CMV in the

Table 2. Reliabilities and average variance extracted (AVE)

Variable	Item	Composite reliability	AVE	Cronbach's alpha
Commitment	3	0.908	0.767	0.847
Reciprocity	2	0.943	0.892	0.878
Self-rated expertise	1	1	1	–
Tenure in the OSS community	1	1	1	–
Centrality	1	1	1	–
Expertise integration	4	0.910	0.716	0.867
Team size	1	1	1	–
Team effort	1	1	1	–
Task completion	2	0.967	0.936	0.931
Technical complexity	3	0.845	0.735	0.647
Task interdependence	4	0.924	0.754	0.894

OSS, open source software.

Table 3. Correlation between constructs

	Mean	SD	TC	TS	TE	Expl	Com	Rec	Exp	Ten	Cen	TecC	TI
TC	3.52	1.17	**0.96**										
TS	3.51	1.14	0.59	**1.00**									
TE	3.54	1.16	0.73	0.50	**1.00**								
Expl	3.58	0.82	0.28	0.25	0.38	**0.84**							
Com	3.52	0.89	0.24	0.15	0.31	0.59	**0.87**						
Rec	3.72	0.88	0.16	0.02	0.22	0.70	0.60	**0.94**					
Exp	1.87	0.86	0.41	0.28	0.34	0.12	0.18	0.05	**1.00**				
Ten	3.48	1.11	0.78	0.53	0.77	0.30	0.26	0.13	0.47	**1.00**			
Cen	2.33	0.96	0.51	0.40	0.36	0.19	0.25	0.07	0.41	0.474	**1.00**		
TecC	3.56	0.87	0.26	0.20	0.30	0.69	0.66	0.57	0.08	0.291	0.05	**0.85**	
TI	3.01	1.00	0.13	–0.11	0.22	–0.19	–0.19	–0.16	0.12	0.13	0.04	–0.23	**0.86**

The numbers in the diagonal row are square roots of the average variance extracted.

SD, standard deviation; Com, commitment; Rec, reciprocity; Exp, self-rated expertise; Ten, tenure in the open source software community; Cen, centrality; Expl, expertise integration; TS, team size; TE, team effort; TC, task completion; TecC, technical complexity; TI, task interdependence.

data set. According to Harman's one-factor test, CMV is high provided that a single factor accounts for a majority of covariance in the independent and dependent variables. Our factor analysis did not detect such a single factor explaining a majority of the covariance. Using the above methods, we believe that CMV is unlikely to occur in this study.

4.2 Structural model

4.2.1 Direct model

The test of the structural model includes assessing the path coefficients (β), which refer to the strengths of the relationships between the dependent and independent variables and the R^2 value, which represents the amount of variance explained by the independent variables and the predictive power of the model. The interpretation of R^2 is the same as that in multiple regressions. This study used bootstrap resampling procedures to generate *t*-statistics, standard error and a confidence estimation procedure (Chin *et al.*, 1998). Resamples of 500 is chosen. Table 4 illustrates path coefficients (β), *t*-value, R^2 and other related data.

Direct model includes the relationships between expertise integration and the effectiveness of an OSS team (Hypotheses 1a, 1b, 2a, 2b and 3), and the relationships between social capital and expertise integration (Hypotheses 4a, 4b, 5a, 5b and 6). H1a, H1b, H2a, H2b are supported, whereas we find no evidence of H3, surprisingly. These findings show that OSS output (i.e. task completion) is affected directly by input to an OSS team – in terms of team size and team effort – and their path coefficients are 0.31 ($p < 0.01$) and 0.56 ($p < 0.01$), respectively. Expertise integration is associated with both team size ($\beta = 0.17$, $p < 0.1$) and team effort ($\beta = 0.35$, $p < 0.01$) significantly, but task completion is not affected by expertise

Table 4. Hypothesis testing

Independent variable → dependent variable	Model 1	Model 2	Model 3	Model 4
Team size → task completion (H1a)	0.31***	0.31***	0.31***	0.32***
	(4.28)	(4.44)	(4.53)	(4.84)
Team effort → task completion (H1b)	0.56***	0.56***	0.56***	0.53***
	(6.74)	(7.00)	(6.82)	(6.48)
Expertise integration → team size (H2a)	0.17*	0.17*	0.17*	0.17*
	(2.20)	(2.18)	(2.07)	(2.20)
Expertise integration → team effort (H2b)	0.35***	0.39***	0.35***	0.35***
	(4.762)	(4.814)	(4.66)	(4.84)
Expertise integration → task completion (TC) (H3)	–0.01	–0.01	–0.01	–0.02
	(–0.17)	(–0.18)	(–0.16)	(–0.62)
Commitment → expertise integration (ExpI) (H4a)	0.01	0.01	0.02	0.01
	(0.21)	(0.17)	(0.30)	(0.21)
Reciprocity → expertise integration (H4b)	0.45***	0.45***	0.41***	0.45***
	(4.82)	(4.83)	(4.84)	(5.11)
Expertise (Exp) → expertise integration (H5a)	–0.01	–0.02	–0.01	–0.01
	(–0.41)	(–0.443)	(–0.36)	(–0.43)
Tenure (Ten) → expertise integration (H5b)	0.08	0.08	0.14*	0.08
	(1.30)	(1.30)	(1.85)	(1.270)
Centrality → expertise integration (H6)	0.10*	0.10*	0.06	0.10*
	(1.75)	(1.73)	(1.21)	(1.72)
Exp * technical complexity → ExpI (H7a)		0.01		
		(0.19)		
Ten * technical complexity → ExpI (H7b)			–0.22***	
			(–3.45)	
ExpI * task interdependence → TC (H8)				–0.11*
				(–2.08)
R^2 (expertise integration)	0.645	0.645	0.685	
R^2 (team size)	0.130			
R^2 (team effort)	0.348			
R^2 (task completion)	0.612			0.642
$R_2^2 - R_1^2$		0	0.040	0.030
f^2		0	0.127	0.110
Test of differenced R^2		0	12.954***	8.652***

Note: $F_{(0.1,1.100)} = 2.756$; $F_{(0.05,1.100)} = 3.936$; $F_{(0.01,1.100)} = 6.895$.
*$p < 0.1$; **$p < 0.05$; ***$p < 0.01$; with one-tailed test.

integration directly (i.e. β = –0.01) – rather, expertise integration influences task completion indirectly through either team size or team effort.

The findings of the relationships between social capital and expertise integration are mixed – H4b and H6 are supported as expected, while surprisingly, H4a, H5a and H5b are not supported. Expertise integration is influenced by both reciprocity (β = 0.45, $p < 0.01$) and centrality (β = 0.10, $p < 0.1$) as expected, it is not affected by OSS members' commitment (H4a, β = 0.01), expertise (H5a, β = –0.01) or tenure (H5b, β = 0.08) significantly.

4.2.2 Moderating effect

Moderating effects can be assured by comparing the difference between the main effect and the moderating effect models (Carte & Russell, 2003; Wang *et al.*, 2006). We first obtained the R-square (R_1^2) of the main effect model, which consists of the independent variable, moderator and dependent variable only. Then, the R-square (R_2^2) of the moderating effect model was estimated by including the independent variable, moderator, interaction term and dependent variable in the model. The interaction terms were calculated by adding the product of each indicator in the independent variable and each indicator in the moderator. We then derived an estimated effect size of f^2 from $(R_2^2 - R_1^2) / (1 - R_1^2)$ and then obtained a pseudo F-value by multiplying f^2 with $(n - k - 1)$, where n is the sample size and k is the number of independent variables in the regression equation. f^2 scores of 0.03, 0.15 and 0.35 indicate small, moderate and large interaction effects, respectively. Finally, we compare the pseudo F-value with $F_{1,n-k-1}$. Based on the above steps, the change of variance extracted by adding a new variable (the interaction term) into the model can be examined. From Table 4, H7b and H8 were supported as seen in models 3 and 4, respectively; H7a was not (model 2).

5. DISCUSSION AND IMPLICATIONS

5.1 The relationship between social capital and expertise integration

This study aims to investigate what factors affect the effectiveness of OSS development, particularly focusing on social capital and expertise integration in loosely knitted communities (OSS communities). Both the theoretical model and most of the proposed hypotheses are supported by our empirical findings. We found that social capital plays an important role in underlying expertise integration despite the weak ties between members and the media richness limitations inherent in online communication. Structural social capital affects expertise integration significantly. Consistent with theories of collective action (Burt, 1992), individuals who are central to the network and connected to a large number of others are more likely to sustain collective actions such as contributing knowledge to the collective and building on each other's expertise during the OSS process (Bergquist & Ljungberg, 2001; Tiwana & McLean, 2005).

Regarding relational capital, the results indicate that reciprocity plays a key role in facilitating expertise integration, but commitment does not. Consistent with prior work (Shumaker & Brownell, 1984; Tiwana & McLean, 2005; Scacchi *et al.*, 2006), reciprocity is critical for sustaining supportive relationships and collective action such as contributing knowledge to other OSS members and building on each other's perspectives and expertise. When there is a strong norm of reciprocity in the collective, knowledge sharing and expertise integration become easier because members have a strong sense of fairness (favours given and received) and trust each other, which in turn increase their willingness to build on each other's perspectives and expertise (Wasko & Faraj, 2002; Tiwana & Mclean, 2005). Contrary to expectations, our results show that commitment does not affect expertise integration significantly. One possible explanation is that unlike generalized network-based interactions, OSS developers' interactions emphasize personal exchanges between two individuals (or dyadic interactions) where there is an expectation of direct reciprocity. Given commitment to a collective is usually affected by a sense of obligation to the organization (Constant *et al.*, 1996), it may not accrue from dyadic interactions. If commitment is not key to facilitating expertise integration in an OSS community, one potentially exciting area of future research would be to apply social network analysis techniques to investigating whether patterns of dyadic interactions substitute for commitment and how.

Regarding cognitive social capital, our findings show that neither tenure nor self-rated expertise is associated with expertise integration significantly. This result is at variance with prior research, which suggests that both individual expertise and tenure are important predicators of knowledge contribution and the helpfulness of replies in electronic networks of practice (Constant *et al.*, 1996; Wasko & Faraj, 2002). There are two possible interpretations for this inconsistency. First, the more expertise an individual developer has, the more likely she has a deep and unique understanding of technical and application domain knowledge, indicating more diverse interpretations of project goals and more different perspectives on possible solutions. This in turn impedes the team's ability to reach consensus on project goals and priorities. This interpretation is consistent with prior studies that increased expertise heterogeneity in OSS groups can decrease expertise integration (Sharma *et al.*, 2002; Tiwana & McLean, 2005).

Second, prior work is not completely applicable to the context of OSS development. The existing body of research that argues a positive relationship between individually held expertise and team processes focuses largely on industrial teams such as those found in assembly lines and manufacturing plants (e.g. Lovelace *et al.*, 2001). The tasks of such teams are less knowledge intensive, entailing lower levels of expertise interdependence (i.e. they rely on simple pooling or sequential application of individual expertise). In contrast, developing an OSS project is more contingent on integrating the expertise of various members (Stewart and Gosain, 2006).

5.2 The relationship between expertise integration and the effectiveness of OSS

One conclusion that can be drawn from this study is that expertise integration is the main driver of the input effectiveness of OSS teams in both size and effort, which in turn jointly affect output effectiveness. The results are consistent with prior research on the relationship between expertise integration and team productivity in an organizational context (Tiwana & McLean, 2005), providing additional evidence that while output effectiveness is not directly affected by expertise integration, it is indirectly associated with output effectiveness through input effectiveness in an OSS community. This may indicate that expertise integration alone is not enough to improve task completion. Rather, its effectiveness is mainly determined by a large pool of developers and those who are willing to devote their time and effort to OSS development.

5.3 The moderating role of technical complexity and task interdependence

As hypothesized, technical complexity negatively moderates the relationship between tenure and expertise integration, and expertise integration is negatively related to task completion contingent on task interdependence. By contrast, we found that the relationship between members' expertise and expertise integration is not contingent on technical complexity. One possible explanation is that given the complexity of an OSS project, fewer developers hold the required expertise, which in turn lowers the variety of ideas and the range of possible linkages and associations among those ideas. Thus, the efforts required for synthesizing members' expertise may not be salient to OSS managers.

5.4 Implications for practice

The results of this study have interesting implications for practitioners interested in OSS development and how to leverage the social capital for competitive advantage. Individuals in an OSS team benefit from accessing external knowledge and building on each other's expertise during the OSS development because valuable expertise flows into the organization at relatively little cost. By participating in an OSS community, individuals gain reputation and become central to a large network of resources. Disallowing such participation may cut off valuable knowledge flows and reduce employee efficacy (Wasko & Faraj, 2005; Scacchi *et al.*, 2006).

Given the effectiveness of OSS development relies directly on how an OSS team integrates members' knowledge at the project level in formulating project solutions, managers interested in developing and sustaining expertise integration through OSS communities should focus attention on the creation and maintenance of a sense of reciprocity. Research (Nahapiet & Ghoshal, 1998) shows that a norm of reciprocity may not develop in a loosely knit group due to a lack of high interdependence and frequent interaction. We therefore suggest that facilitating collaboration becomes an important issue to be addressed. Managers should focus on building a strong sense of reciprocity and fairness to help foster collaboration.

Leveraging centrality and promoting individual reputations may also help signal the potential quality of responses to novice participants and lurkers, making the knowledge more accessible to all participants in the network, which in turn may facilitate the expertise integration. As suggested by von Hipple & von Krogh (2003), techniques that identify a member's centrality can effectively support knowledge sharing and integration by helping knowledge seekers evaluate the quality of responses to their questions and by finding linkages among the diverse perspectives and domain expertise. Gaining status and recognition in this way are likely to motivate OSS developers to participate more in the OSS community (Wasko & Faraj, 2005; Scacchi *et al.*, 2006). Thus, making centrality a part of an individual's identification may provide an incentive for members to respond frequently and helpfully to other members.

5.5 Implications for research

Because of the explosion in OSS usage and development in the last decade, exploration of how to facilitate the effectiveness of OSS is worthwhile, particularly focusing on the role of social capital and expertise integration. How social capital affects expertise integration, which in turn influences the effectiveness of OSS in an electronic network of practice setting, will be critical to companies in the future. Extending prior word, this is the first empirical study aimed at identifying the relationship between social capital and effectiveness of free OSS development based on sound theories – social capital theory and a knowledge-based view of organizing a free OSS development team (Bergquist & Ljungberg, 2001; Wasko & Faraj, 2005; Scacchi *et al.*, 2006). Social capital theory is a theoretically rich model because it aims to explain collective action in a comprehensive way. Thus, applying it to the context of free OSS development extends our understanding of how social capital facilitates the integration of OSS team members' expertise. Scacchi *et al.*'s (2006) study shows that free OSS developers interconnecting multiple free OSS projects is a way to accumulate social capital. Combining their arguments and our findings, we argue that social capital facilitates OSS development and knowledge contributions, which in turn lead to more accumulation of social capital in the community. Future research may focus on how to facilitate the establishment of social capital in OSS communities. Furthermore, this study also considers the difference between traditional IS and free OSS development, in terms of governance and development processes, and our model is contingent on technical complexity and task interdependence.

Our model can easily be expanded to include other related issues. For example, developing a private-collective model of innovation incentives (von Krogh & Spaeth, 2007), which suggests that developers contribute to public good innovation because they garner private benefits related to the innovation process. These benefits including fun, reputation and peer recognition are not supplied to the same degree to non-contributors. In addition, the culture of the free OSS development and the corresponding social norms and beliefs (or OSS ideology) that regulate the behaviour of an OSS community's members also deserve further investigation because the above issues are significantly different from those of traditional ISD.

5.6 Limitations and future study

This study has some limitations. First, while we aim to investigate the causal nature of the relationships in online OSS communities, the research methodology (i.e. a cross-sectional survey) does not allow us to establish a longitudinal study. Future study may focus on how the relationship between expertise integration and task completion is affected by trust because trust has generally been argued to develop over time (Stewart & Gosain, 2006). Second, the measure of task completion overcomes known problems with other indicators of IS performance such as lines of code (von Hipple & von Krogh, 2003). However, this, together with team size and team effort, did not take into account any specific evaluation of quality. For example, it is possible that teams could report tasks as having been completed without an appropriate manner. Besides, given that task identification and completion may follow cycles related to new software releases, it is likely the timing of data collection plays a key role. Future efforts may focus on seeking more discriminating measures of OSS team effectiveness and including more dynamic models to capture effects of timing of key events in projects. Third, as several factors may affect the performance of OSS (in terms of team size, team effort and task completion), such as how comfortably are the developers employed in other companies or business, what sort of profile the project has in both the media and in hacker eyes, and so on, future study may address their impact on an OSS's performance. Finally, as mentioned in Section 3, using OSS project administrators as key informants could be a drawback of this study. Future study may consider the proposed model from other perspectives such as individual OSS developers' viewpoint.

6. CONCLUSIONS

The emerging work on understanding OSS argued for the importance of exploring what are the determinants that affect the effectiveness of free OSS development. This is so because hundreds of free OSS systems are now in widespread use by many end-users and the principles and practices used in traditional ISD cannot be applied directly to the context of free OSS development (Scacchi *et al.*, 2006). Building social relationships in open source communities and leveraging social capital have been recognized as key factors that affect the development of a free OSS project (Bergquist & Ljungberg, 2001; von Hipple & von Krogh, 2003; Tiwana & McLean, 2005; Wasko & Faraj, 2005). As a systematic empirical study based on social capital (in terms of relational, cognitive and structural capital) theory and a knowledge-based view of organizing a free OSS development team (in terms of governance and development processes) remains absent, this study develops and validates a theoretical model drawing on the above theories. In addition, we also examined whether our model is contingent on key antecedents (technical complexity and task interdependence), as they play a key role in traditional ISD. Based on 160 useful respondents, most of the proposed hypotheses were supported. Reciprocity and centrality exerted positive impact on expertise integration, while the influences of commitment and members' cognitive capital were insignificant. Besides, the relationship between tenure and expertise integration is negatively contingent on technical complexity, and task interdependence adversely moderates the relationship between expertise integration and task completion. We hope that these findings encourage other researchers to delve more deeply into the varying roles that social capital plays in effectiveness of free OSS development.

REFERENCES

Ahuja, M., Galletta, D. & Carley, K. (2003) Individual centrality and performance in virtual R&D groups: an empirical study. *Management Science*, **49**, 21–38.

Bergquist, M. & Ljungberg, J. (2001) The power of gifts: organizing social relationships in open source communities. *Information Systems Journal*, **11**, 305–320.

Blau, P.M. (1964) *Exchange and Power in Social Life*. John Wiley, New York, NY, USA.

Brown, J.S. & Duguid, P. (1991) Organizational learning and communities-of-practices: toward a unified view of working, learning, and innovation. *Organization Science*, **2**, 40–57.

Burt, R.S. (1992) *Structural Hole: The Social Structure of Competition*. Harvard Business School Press, Cambridge, MA, USA.

Carte, T.A. & Russell, C.J. (2003) In pursuit of moderation: nine common errors and their solutions. *MIS Quarterly*, **27**, 479–501.

Chin, W.W., Marcolin, B.L. & Newsted, P.R. (1998) A partial least squares latent variable modeling approach for measuring interaction effects: results from a Monte Carlo simulation study and an electronic-mail emotion/adoption study. *Information Systems Research*, **14**, 189–217.

Choi, I. & Kim, Y.G. (2002) Critical factors for assimilation of object-oriented programming languages. *Journal of MIS*, **18**, 125–156.

Coleman, J.S. (1990) *Foundations of Social Theory*, Belknap Press, Cambridge, MA, USA.

Constant, D., Sproull, L. & Kiesler, S. (1996) The kindness of strangers: the usefulness of electronic weak ties for technical advice. *Organization Science*, **7**, 119–135.

Cooper, R. (2000) Information technology development creativity: a case study of attempted radical change. *MIS Quarterly*, **24**, 245–276.

Crowston, K., Annabi, H. & Howison, J. (2003) Defining open source software project success. In: *Proceedings of the 24th International Conference on Information Systems*, March, S.T., Massey, A. & DeGross, J.I. (eds), pp. 327–340. Seattle, WA, USA.

Edmondson, A.C., Bohmer, R.M. & Pisano, G.P. (2001) Disrupted routines: team learning and new technology implementation in hospitals. *Administrative Science Quarterly*, **46**, 685–716.

Ethiraj, S.K., Kale, P., Krishnan, M.S. & Singh, J.V. (2005) Where do capabilities come from and how do they matter? A study in the software services industry. *Strategic Management Journal*, **26**, 25–45.

Fichman, R.G. & Kemerer, C.F. (1997) The assimilation of software process innovations: an organizational learning perspective. *Management Science*, **43**, 1345–1363.

Fornell, C. & Larcker, D.F. (1981) Structural equation models with unobservable variables and measurement errors. *Journal of Marketing Research*, **18**, 39–50.

Gallivan, M.J., Spitler, V.K. & Koufaris, M. (2005) Does information technology training really matter? A social information processing analysis of coworkers' influence on IT usage in the workplace. *Journal of MIS*, **22**, 153–192.

Grant, R. (1996) Prospering in dynamically-competitive environments: organizational capability as knowledge integration. *Organization Science*, **7**, 375–387.

Hackathorn, R.D. & Keen, P.G.W. (1981) Organizational strategies for personal computing in decision support systems. *MIS Quarterly*, **5**, 21–27.

Hair, J.F., Andeson, R.E., Tatham, R.L. & Black, W.C. (1998) *Multivariate Data Analysis*, 5th edn. Prentice Hall, Englewood Cliffs, NJ, USA.

Herbsleb, J.D. & Mockus, A. (2003) An empirical study of speed and communication in globally-distributed software development. *IEEE Transactions on Software Engineering*, **29**, 1–14.

Hickey, A. & Davis, A. (2004) A unified model of requirements elicitation. *Journal of MIS*, **20**, 65–84.

von Hipple, E. & von Krogh, G. (2003) Open source software and the private-collective innovation model: issues for organization science. *Organization Science*, **14**, 209–223.

Ilgen, D.R., Hollenbeck, J.R., Johnson, M. & Jundt, D. (2005) Teams in organizations: from input-process-output models to IMOI models. *Annual Review of Psychology*, **56**, 517–543.

Karimi, J., Somers, T.M. & Gupta, Y.P. (2004) Impact of environmental uncertainty and task characteristics on user satisfaction with data. *Information Systems Research*, **15**, 175–193.

von Krogh, G. & Spaeth, S. (2007) The open source software phenomenon: characteristics that promote research. *Journal of Strategic Information Systems*, **16**, 236–253.

Lee, S.Y.T., Kim, H.W. & Gupta, S. (2009) Measuring open source software success. *Omega*, **37**, 426–438.

Lin, N. (2001) *Social Capital*. Cambridge University Press, Cambridge, UK.

Lovelace, K., Shapiro, D. & Weingart, L. (2001) Maximizing cross-functional new product teams' innovativeness and constraint adherence: a conflict communications perspective. *Academy of Management Journal*, **44**, 779–793.

Madey, G., Freeh, V. & Tynan, R. (2004) Modeling the F/OSS community: a quantitative investigation. In: *Free/Open Source Software Development*, Koch, S. (ed.), pp. 203–221. Idea Group Publishing, Hershey, PA, USA.

Markus, M.L., Manvile, B. & Agres, C.E. (2000) What makes a virtual organization work? *Sloan Management Review*, **42**, 13–26.

Marwell, G. & Oliver, P. (1993) *The Critical Mass in Collective Action: A Micro-Social Theory*. Cambridge University Press, New York, NY, USA.

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

Mowday, R.T., Steers, R.M. & Porter, L. (1979) The measurement of organizational commitment. *Journal of Vocational Behavior*, **14**, 224–247.

Nahapiet, J. & Ghoshal, S. (1998) Social capital, intellectual capital, and the organizational advantage. *Academy of Management Review*, **23**, 242–266.

Napier, N.P., Keil, M. & Tan, F.B. (2009) IT project managers' construction of successful project management practice: a repertory grid investigation. *Information Systems Journal*, **19**, 255–282.

Okhuysen, G. & Eisenhardt, K. (2002) Integrating knowledge in groups: how formal interventions enable flexibility. *Organization Science*, **13**, 370–386.

Pearce, J.L., Sommer, S.M., Morris, A. & Frideger, M. (1992) A configurational approach to interpersonal relations: profiles of workplace social relations and task interdependence. GSM#OB92015 1992. University of California, Irvine, CA, USA.

Robey, D., Ross, J.W. & Boudreau, M.C. (2002) Learning to implement enterprise systems: an exploratory study of the dialectics of change. *Journal of MIS*, **19**, 17–46.

Rogers, E.M. (1983) *Diffusion of Innovations*, 3rd edn. The Free Press, New York, NY, USA.

Scacchi, W. (2002) Understanding the requirements for developing open source software systems. *IEE Proceedings on Software*, **149**, 24–39.

Scacchi, W., Feller, J., Fitzgerald, B., Hissam, S. & Lakhani, K. (2006) Understanding free/open source software development processes. *Software Process Improvement Practice*, **11**, 95–105.

Sharma, R. & Yetton, P. (2007) The contingent effects of training, technical complexity, and task interdependence on successful information systems implementation. *MIS Quarterly*, **31**, 219–238.

Sharma, S., Sugumaran, V. & Rajagopalan, B. (2002) A framework for creating hybrid-open source software communities. *Information Systems Journal*, **12**, 7–25.

Shumaker, S. & Brownell, A. (1984) Toward a theory of social support: closing conceptual gaps. *Journal of Social Issues*, **40**, 11–36.

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**, 316–325.

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

Tiwana, A. & McLean, E. (2005) Expertise integration and creativity in information systems development. *Journal of MIS*, **22**, 13–43.

Wang, E.T.G., Klein, G. & Jiang, J.J. (2006) ERP misfit: country of origin and organizational factors. *Journal of MIS*, **23**, 263–292.

Wasko, M. & Faraj, S. (2002) It is what one does: why people participate and help others in electronic communities of practice. *Journal of Strategic Information Systems*, **9**, 155–173.

Wasko, M. & Faraj, S. (2005) Why should I share? Examining social capital and knowledge contribution in electronic networks of practice. *MIS Quarterly*, **29**, 35–57.

Wu, C.G., Gerlach, J.H. & Young, C.E. (2007) An empirical analysis of open software developers' motivations and continuance intentions. *Information and Management*, **44**, 253–262.

Biographies

Shih-Wei Chou is a Professor at the department of management of information systems (MIS) at National Kaohsiung First University of Science and Technology in Taiwan where he teaches knowledge management, software engineering, object-oriented information systems (IS) and management of IS. He received his doctorate in computer science (CS) from Illinois Institute of Technology, and he holds a master's degree in CS from Mississippi State University. Prior to getting his doctorate, he spent 2 years working in the telecommunications industry. His research interests include knowledge management, IS development and design, enterprise resource planning systems, Information Technology behaviour. His work has appeared in the *International Journal of Human-Computer Studies*, *Journal of Information Science*, *Journal of Computer-Aided Learning*, *Decision Support Systems*, and *Computers & Education*, and has been presented at the Pacific Asia Conference on Information Systems, Hawaii International Conference on Systems Science and other international conferences.

Mong-Young He is currently a PhD candidate at the MIS department of National Kaohsiung First University of Science and Technology in Taiwan. His research interests include information systems and open source software development, electronic commerce and knowledge management.