View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by DCU Online Research Access Service

Ryan, S. and O'Connor, R., Acquiring and Sharing Tacit Knowledge in Software Development Teams: An Empirical Study, Information and Software Technology, Vol. 55, No. 9, pp. 1614 - 1624, 2013.

Acquiring and Sharing Tacit Knowledge in Software Development Teams: An Empirical Study

Sharon Ryan[a], Rory V. O'Connor[b]

[a] *RepGraph, UK*
sryan@repgraph.co.uk
[b] *School of Computing, Dublin City University and Lero, Irish Software Engineering Research Centre, Ireland*
roconnor@computing.dcu.ie

ABSTRACT

Context: Sharing expert knowledge is a key process in developing software products. Since expert knowledge is mostly tacit, the acquisition and sharing of tacit knowledge along with the development of a transactive memory system (TMS) are significant factors in effective software teams.
Objective: We seek to enhance our understanding human factors in the software development process and provide support for the agile approach, particularly in its advocacy of social interaction, by answering two questions*: How do software development teams acquire and share tacit knowledge*? *What roles do tacit knowledge and transactive memory play in successful team performance*?
Method: A theoretical model describing the process for acquiring and sharing tacit knowledge and development of a TMS through social interaction is presented and a second predictive model addresses the two research questions above. The elements of the predictive model and other demographic variables were incorporated into a larger online survey for software development teams, completed by 46 software SMEs, consisting of 181 individual team members.
Results: Our results show that team tacit knowledge is acquired and shared directly through good quality social interactions and through the development of a TMS with quality of social interaction playing a greater role than transactive memory. Both TMS and team tacit knowledge predict effectiveness but not efficiency in software teams.
Conclusion: It is concluded that TMS and team tacit knowledge can differentiate between low- and high-performing teams in terms of effectiveness, where more effective teams have a competitive advantage in developing new products and bringing them to market. As face-to-face social interaction is key, collocated, functionally rich, domain expert teams are advocated rather than distributed teams, though arguably the team manager may be in a separate geographic location provided that there is frequent communication and effective use of issue tracking tools as in agile teams.

Key Words: Tacit knowledge, Transactive memory, Social interaction, Agile teams, Team performance

1. INTRODUCTION

Various observers describe today's global economy as one in transition to a 'knowledge economy' as an extension of an 'information society'. As the modern world economy has transitioned from an industrial to knowledge-based economy, the nature of software development has also changed. Software development is a knowledge-intensive process, where knowledge is created and shared, when different aspects of a software development process (concepts, products, tools, process, people etc.) interact with each other [1]. Software development is a knowledge-driven industry, which relies on employees' expert knowledge to create a finished product, where this expert knowledge is mostly tacit [2] Software development teams have a relatively unique structure, wherein the division of labour among members is highly interdependent. Therefore knowledge sharing is a key process in developing software products and since expert knowledge is mostly tacit, the acquisition and sharing of tacit knowledge and the development of a Transactive Memory System (TMS) are significant in the software development process.

Tacit knowledge, as opposed to formal or explicit knowledge, refers to a category of knowledge that is difficult to transfer to another person by means of writing it down or verbalising it. The concept of tacit knowledge was introduced by Polanyi [3] who described it as knowledge that cannot be articulated, or the fact that "*we can know more than we can tell*" (p.4).

Since then, there has been much debate in the literature as to how tacit knowledge can be conceptualised and operationally defined. Some researchers [4] argue that 'tacitness…is a matter of degree' and that the same knowledge may be more tacit for one person than p.78. Others argue that there is a middle ground between tacit and explicit knowledge, which is articulable tacit knowledge [2, 5, 6, 7]. This is where tacit knowledge is created through social interaction [2, 7]. Yet others dispute that tacit knowledge can ever be articulated [3, 8] and refer to this middle ground as implicit knowledge.

Domain knowledge required by experts, such as software developers, to perform expert roles in organisational contexts is largely tacit and this underlines the need for knowledge sharing to enable software organisations to:

- Effectively share domain expertise between the customer and the development team;
- Identify the requirements of the software system;
- Capture non-externalised knowledge of the development team members;
- Bring together knowledge from distributed individuals to form a repository of organisational knowledge;
- Retain knowledge that would otherwise be lost due to the loss of experienced staff; and
- Improve organisational knowledge dissemination [9]

There are multiple approaches to organizing the software development process and multiple factors influencing the software development process [10], with two major ones being the traditional (or plan based), which rely primarily on managing explicit knowledge, and agile methods, which primarily rely on managing tacit knowledge [11] and recognises the importance of human interaction in the software process over written knowledge in formal documentation (see http://agilemanifesto.org). Communication in agile development is both crucial but also tacit, informal and predominantly verbal [12]. Understanding the role and nature of tacit knowledge in agile software development is challenging given agile's discounting of comprehensive documentation and its value on human interaction.

The increasing popularity of agile methods makes it important to test its underlying principles and thereby enhance our understanding human factors in the software development process. This will serve to provide support for the agile approach, particularly in its advocacy of social interaction. Accordingly this paper addresses the following questions:

1. How do software development teams acquire and share tacit knowledge?
2. What roles do tacit knowledge and transactive memory play in successful team performance?

In the present study, a novel measure of team transactive memory is introduced as a part of a predictive model exploring the nature of the acquisition and sharing of tacit knowledge in software development teams. A transactive memory system (TMS) is a type of shared mental model, where there is a cooperative division of labour for learning, remembering, and communicating relevant team knowledge. Building on an earlier study [2] which developed a team measure for tacit knowledge in software development teams, we now include both quality and quantity of social interaction as part of the predictive model. This model is expected to provide empirical support for the agile approach to software development. Among other guiding principles, agile methods emphasise the importance of tacit knowledge sharing through social interaction. While recent studies have emerged which support the link between tacit knowledge and social interaction in agile teams, there is a clear lack lack of statistical and empirical studies to support agile claims [13]. In addition, there is a lack of rigour in the measures for tacit knowledge and a paucity of evidence at the team level as to how this knowledge is actually integrated into the team and subsequently manifest in performance. It is proposed in this research that social interaction is central to development of a TMS in software teams and also to tacit knowledge creation. This TMS assists software development teams in acquiring and disseminating tacit knowledge. Furthermore, the predictive model also explores the impact of transactive memory, social interaction and tacit knowledge on team performance.

2. BACKGROUND

In a previous study [2] we operationally defined and developed a measure of Team Tacit Knowledge (TTK) for software development teams and demonstrated that team tacit knowledge was positively related to good quality social interactions and predicted effective team performance. Furthermore we demonstrated that tacit knowledge is acquired and shared directly, through good quality social interactions and the frequency of interaction indirectly aids the acquisition and sharing of tacit knowledge since it leads to better quality interactions.

The development and validation of the Team Tacit Knowledge Measure (TTKM) for software development teams was an extension of individual-level, tacit-knowledge research to consider team-level behaviour. In this paper, team tacit knowledge is defined as: *The aggregation of articulable tacit, individual, goal driven, expert knowledge to the team-level, where different members of the team possess different aspects of tacit knowledge* [2]

2.1. Transactive Memory and Social Interaction

Transactive memory systems (TMSs) were conceived by Wegner [14] who observed that members of long-tenured groups tend to rely on one another to obtain, process, and communicate information from distinct knowledge domains. Wegner [14] posited that knowledge specialisation is greater in groups with well-developed TMSs. Specialisation involves expertise recognition, as it guides group members to identify other team members with the relevant information and evaluate that information based on the source [15]. Transactive memory involves the awareness of specialisations (or expert knowledge) and coordination of this differentiated knowledge. Specialised knowledge and its coordination may be acquired through experience of working in a domain, in this case software development.

Lewis [16] developed a field measure of TMSs, holding that a TMS was an unobservable (latent) construct which could be discerned from the following manifest variables: the differentiated structure of members' knowledge (specialisation), members' beliefs about the reliability of other members' knowledge (credibility), and effective orchestrated knowledge processing (coordination). According to Lewis and Herndon [17] the formal assumptions which underlie the latent variable model is that (a) we *infer* that TMS is operating in a team through the manifest variables of credibility, specialisation and coordination (b) these variables co-vary as a function of TMS and (c) the manifest variables are independent after controlling for a TMS, i.e. there is no other reason for the three manifest variables to be related.

2.2 Social Interaction and the development of TMS

In the present study, social interaction is defined as face-to-face, tied to a goal and informal where good quality of social interactions are related to tacit knowledge. The quality of social interaction is an important means through which tacit knowledge is created in teams, and it is now proposed that social interaction is central to the development of a TMS [2]. Social interaction in teams is related to 'Team Mental Models' (TMM), where team members tend to rely on one another in a cognitively interdependent manner. Theory and research surrounding the concept of social interaction do not tend to discriminate between quality and quantity of interaction and definitions tend to be broad rather than focused. Indeed, most definitions use social interaction and communication interchangeably [18, 19]. Quality and quantity of social interaction are interdependent the quality of communication within a team depends on frequency, formalisation, structure and openness of the information exchange [2, 20].

Transactive Memory is a form of team mental model and as such it is developed through social interaction within the team, where informal interaction is considered the most successful type of communication in groups. TMSs develop as team members learn about one another's expertise [14], accomplished predominantly through interpersonal communication [21]. Evidence for the relationship between transactive memory and social interaction is found in the field study by Lewis [16] who measured functional or 'task-relevant' communication and found that it was related to transactive memory. Laboratory studies have also consistently shown TMSs to predict higher performance in work teams [22 ,23] than non-interacting dyads. Furthermore, in a review of the research into TMS [17] cited results from field based studies which found that group member familiarity, communication volume and frequency, and task characteristics of interdependence, cooperative goal interdependence and support for innovation were related to higher TMSs in work groups. However, it must be noted that these studies do not differentiate between quality and quantity of interaction. The nature of the task is also important; according to Akgün [24] the impact of a TMS on team learning, speed-to-market, and new product success was higher when there was a higher task complexity. Software development teams work on complex tasks with many interacting elements which need to be coordinated and integrated.

In a study of 218 individuals across 18 work teams Yuan et al. [25] found that at the individual level the relationship between location of expertise and expertise exchange was mediated by communication tie strength and moderated by shared task interdependence. Team-level variables also were significantly related to individual-level outcomes such that individual expertise exchange happened more frequently in teams with well-developed team-level expertise directories, as well as with higher team communication tie strength and shared task interdependence.

2.3 Sharing Tacit Knowledge

The term knowledge sharing is a more appropriate expression than knowledge transfer and will be used to describe the sharing of knowledge between people with emphasis on knowledge sharing within groups and teams. It is argued that simple knowledge can be shared formally through scheduled meetings, training, lectures and formal discussion. However, sharing complex knowledge, as in software development teams, involves the development of TMMs or a collective mind [7, 26]. The TMM of interest in the present study is transactive memory. Knowledge sharing is argued to be a complex social process that involves eliciting both explicit and tacit knowledge. The process is further complicated by the need to fully understand and consider the context within which the knowledge is embedded [27]. Tacit knowledge, like knowledge in general, may be common to a group or divided over individuals.

3. ACQUIRING AND SHARING TACIT KNOWLEDGE IN TEAMS

Knowledge driven industries such as the software sector rely on employees' expertise to produce a finished product. Knowledge is the means of production in such organisations, and this expert knowledge is owned by employees. Expert knowledge is mainly tacit or inarticulable, and so difficult to communicate however, it is thought to be a core competitive advantage [7, 28, 29, 30]. Social interaction between team members is forwarded as the means through which tacit knowledge is acquired and shared [7, 31, 32]. Software products are developed in teams and it is therefore essential to the effective development of these products that knowledge is acquired and shared within the development team.

Dyer [33] defines a team as 'a collection of people who must collaborate, *to some degree,* to achieve common goals' (pp. 24-25). He goes on to suggest that various types of teams can be placed along a continuum according to the amount of collaboration (integration and role differentiation) required. At one end of the continuum are teams, such as golf teams, that are composed of a set of individual performers. At the other end he places the crew of an Air Force bomber where every member of the crew has a specific set of assignments that are critical if the venture is to be successful. Often times, group and team are used interchangeably [32, 34]. In the present study the unit of analysis is referred to as team rather than group as this implies that there is interaction between members. In addition, software development teams may be placed towards the collaborative end of the continuum.

3.1 Philosophical Underpinning

Whilst a universal definition of knowledge remains elusive, it is necessary for the development of knowledge sharing theories, to have at least a working definition to inform development. Berger and Luckmann [35] forwarded a discourse called 'The Social Construction of Reality' which asserted that knowledge begins with the individual, but through face-to face interaction, a shared reality is constructed that is agreed upon socially and is situation dependent. Knowledge in everyday life is socially distributed being possessed differently by different individuals and types of individuals. For the purpose of this research the philosophical approach taken draws on the phenomenological and constructivist theories of Von Glaserfeld [1, 35, 36]. Phenomenological and constructivist approaches to studying knowledge do not separate the knower from the knowledge they possess and advocate a socially agreed upon reality. This study's philosophical approach to knowledge sees knowledge as personal, socially constructed and rooted in tacit knowledge

3.2 Levels of Knowledge Acquisition

Acquiring tacit knowledge involves organisational learning which occurs at several levels. Edmondson [37] outlined three levels of theorising about organisational learning. At the 'macro' or organisational level, theories focus on the stabilising effects of routines and adaptation over time. Individual level or micro approaches look at the behaviour of individuals and their effect on organisational change. The macro and micro levels of analysis provide a foundation for a third perspective that investigates learning at the group or 'meso' level of analysis, where a group level approach is inherently integrative, incorporating factors from two or more levels simultaneously [38]. According to Edmondson [37] teams, or work groups, are also important in that individual cognition and behaviour, is shaped by social influences, that is, by the attitudes and behaviours of others with whom they work closely [39, 40]

Organisational knowledge processes and organisational learning are interdependent, and it is impossible to study one element without studying the other [41, 29]. Johannessen et al. [41] also posit that situated and contextual learning are the elements that tie tacit knowledge to organisational learning. Individual approaches to the acquisition of tacit knowledge, are concerned with the development of expertise, through perception, intuition and experience, and involves deliberative practice. At the team (or 'meso') level, social knowledge requires mutual adjustment and is rarely reproduced in the same way twice and so defies precise codification [32]. It may therefore be considered as mostly tacit.

In the present study the type of tacit knowledge under scrutiny refers to informal face-to-face interaction requiring social interaction to communicate [7, 42, 43]. Hansen et al. [43] argue that interpersonal, relatively close relationships and personal contact, were imperative in transferring complex knowledge i.e. tacit and context dependent, but not for the transfer of simple knowledge i.e. explicit and context-independent. Tacit knowledge may be transferred in a number of ways, including mentoring and apprenticeships, but mainly involves social interaction [2, 31]

3.3 Linking Social Interaction, Tacit Knowledge and Transactive Memory

Social interaction was seen as necessary for the development of TMSs [16]. Social interaction is also related to tacit knowledge, where face-to-face interaction is considered to be the richest medium for transferring knowledge because it allows for immediate feedback and the embodiment of tacit knowledge cues [44]. Face-to-face conversation is best suited to transmitting knowledge that is fundamentally tacit, because it can use a much wider variety of metaphors than conversation through information technology [45]. Furthermore Granovetter [46] from his studies using Social Network Analysis (SNA) stated that strong ties identified by close relationships (among other things) are ideal for the sharing of tacit, complex knowledge. Nonaka and Takeuchi [7] posited in their SECI (Socialization, Externalization, Combination, and Internalization) model that new knowledge is created through iterative social interaction, where tacit knowledge is made explicit. However, a more appropriate explanation may be that rather than making tacit knowledge explicit through social interaction, evidence of tacit knowledge acquisition may be seen in skilled performance [47]. We propose that tacit knowledge acquisition is a reciprocal process, which originates with individuals and becomes group and organisational knowledge as a result of social interaction [48, 49].

3.4 Empirical Evidence for Transactive Memory in Software Teams

TMSs emphasise members' expertise and mental representations of that expertise, but not other mental representations that team members might share about the team, task, or situation. It is this specific emphasis on expertise however, that makes the TMS construct especially relevant for understanding how knowledge-worker teams develop, share, integrate and leverage distributed expertise [16]. Two studies in the domain of software development into the related concepts of expertise coordination and mutual knowledge provide further evidence for the use and development of transactive memory in software development teams. In the expertise coordination study, Faraj and Sproull [50] found 'that for expertise coordination to be effective, processes that are distributed, heedful and emergent have to occur' (p.1556). An empirical investigation into the similar concept of 'mutual knowledge' was conducted by McChesney and Gallagher [51]. These authors posit that 'mutual knowledge' consists not only of specific pieces of information, but also the awareness that the other knows this information. This view sees the team as a distributed cognitive system, highlighting issues of team design and development. These two concepts are very similar to transactive memory. Furthermore, in both studies the coordination of expertise and mutual knowledge were considered tacit.

3.5 Software Teams and Tacit Knowledge Sharing

The processes of communication, coordination, and collaboration are at the heart of, and key enablers of, software development processes [52]. In agile methods communication is the imparting or interchanging of thoughts, opinions, or information by speech, writing, or signs [53]. Communication is an essential component of all software development coordination and collaboration practices and processes, with face-to-face communication is found to be the most effective in software teams [54, 55]. In support of this the Agile Manifesto calls for collaboration and social interaction, emphasises people over processes, working software over documentation, and adaptability to change more than following a fixed plan [56]. Indeed agile methods suggest that most written documentation can be replaced by informal communications among team members internally and between team and the customers with a stronger emphasis on tacit knowledge rather than explicit knowledge [9].

According to Chau et al. [9] it is unlikely that all members of a development team possess all the knowledge required for the activities of software development. Therefore different people will have different aspects of knowledge, as posited by transactive memory and TMM theory. Melnik and Maurer [57] argued that knowledge is socially constructed and held, and conducted a study to demonstrate the importance of face-to-face interaction in sharing abstract or complex knowledge. The authors concluded that the higher the complexity, the more is the need for interactive knowledge sharing via direct verbal communication, citing the richness of face-to-face communication in providing information through physical cues and voice inflection, which are important when there is ambiguity [57].

In another study, Bahli and Zeid [58] explored knowledge sharing in an eXtreme programming (XP) project and a traditional project and found that the creation of tacit knowledge improved as a result of frequent interactions.

3.6 Role of Tacit Knowledge in Team Performance

Evidence of tacit knowledge acquisition is seen in team performance [47]. Team performance on software development projects is dependent on many different and interacting factors like effective plans, good communication, clear goals etc. In addition internal group processes, particularly those focussing on the team's relationships are more likely than technical factors to be associated with team performance on successful projects [59]. In terms of team performance TMMs are assumed to enhance the quality of teamwork skills and team effectiveness [60, 61].

Team performance in software development may be divided into two parts: efficiency and effectiveness [50, 62, 63]. Efficiency usually refers to the budget and schedule of the project [64]. Effectiveness refers to the achievement of project goals [65]. Unrealistic schedules and underestimated effort estimates typically result in extreme workload pressure [64, 66]. When there is a perception that the schedule or effort estimates are unrealistic, software developers may not strive for quality solutions or may not fully commit to the goals of the project [67].

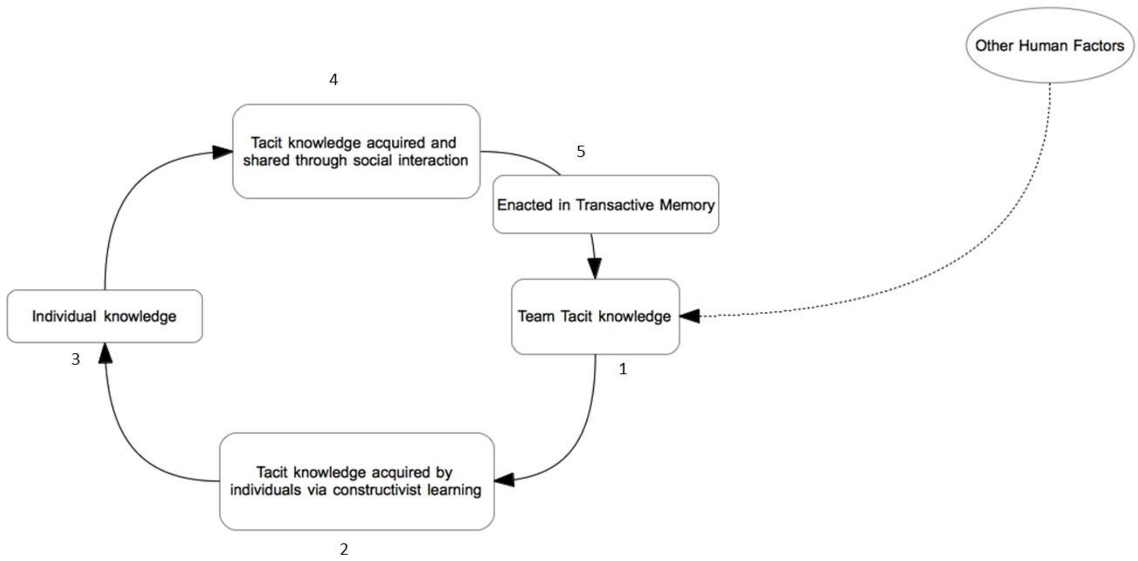
4. MODEL FOR ACQUISITION AND SHARING OF TACIT KNOWLEDGE IN TEAMS

Social interaction is posited as the primary means by which tacit knowledge is shared [2, 7, 32]. Interaction is important because knowledge is stored in communities and groups and a repository on its own does not support these communities [68]. The more social interaction in work groups and teams, the more tacit knowledge is shared, then the better use is made of transactive memory. However it is not merely social interaction alone. Team tacit knowledge refers to knowledge associated with group activities, gained through the experience of working together by constructing shared cognitions or a 'collective mind' [18, 26, 50]. The collective mind is emergent and therefore is not known in its entirety to any one individual although portions are known differentially to all [69]. TMSs enact collective minds of teams.

4.1 Theoretical model for the Acquisition and Sharing of Tacit knowledge in Teams

Knowledge acquisition and sharing are interdependent activities that occur between members of teams and may be seen in their interactions with one-another. Social interactions are therefore essential to the acquisition of team tacit knowledge. Figure 1 illustrates the proposed theoretical link between social interaction, transactive memory and tacit knowledge for the acquisition and sharing of tacit knowledge in teams dealing with complex tasks. It represents the process by which tacit knowledge is shared between individuals in a team and from the team to the individual, where it becomes re-integrated. It is acknowledged that many human factors (e.g. trust, leadership, cohesion etc.) other than social interaction and transactive memory contribute to the creation of team tacit knowledge, these are beyond the scope of the proposed model but their presence is recognised and represented by a dotted line in Figure 1.

Figure 1 Theoretical Model for the Acquisition and Sharing of Tacit Knowledge in Teams



In order to explain the theoretical model each section of the model is numbered and described. This is to ensure clarity; the cognitive processes in the model do not occur in a stepwise sequence but are reciprocal, dynamic and at times simultaneous.

1	Team tacit knowledge has been (and is being) created by team members
2	Individuals draw from the team tacit knowledge and create their own tacit knowledge. This is a background process which is dynamic and reciprocal relying on constructivist situated learning
3	This knowledge is re-integrated and becomes individual knowledge
4 & 5	As individuals interact, informally and face-to-face, tacit knowledge is acquired and shared and a TMS is also developed. TMSs allow for knowledge to be stored and shared, and are therefore both dynamic and static

Transactive processes also refine information in the TMS structure and produce individual and collective learning that is useful to performance which cannot be performed by technology substitutes such as knowledge repositories, intranets, search engines etc. [17].

In a preliminary exploration of this theoretical model the strength of the relationships will be investigated between the main variables in the model. We advance three hypotheses for the acquisition and sharing of tacit knowledge in software teams:

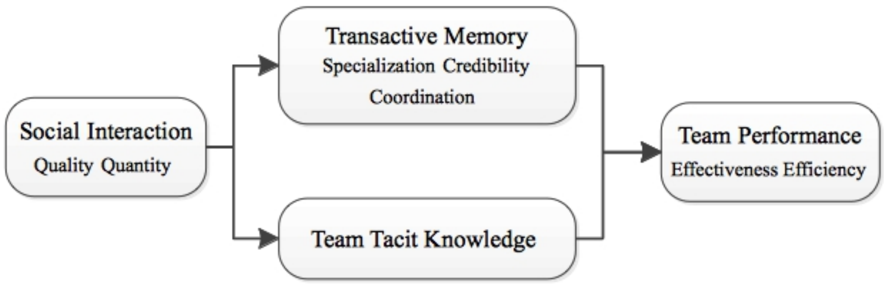
- ***H1**: There will be a positive correlation between social interaction (quality and quantity) and transactive memory*
- ***H2**: There is a positive relationship between social interaction (quality and quantity) and tacit knowledge*
- ***H3**: There will be a positive relationship between transactive memory and tacit knowledge*

4.2 Predictive Model for Acquisition and Sharing of Tacit Knowledge in Teams

In order to further test the theoretical model a second predictive model is forwarded which serves as an operational version of the theoretical model and also includes a prediction of how the variables influence team performance. Transactive memory and tacit knowledge are thought to influence team performance. According to Lewis and Herndon [17] "*(T)he dynamic interplay between TMS structure and processes can produce individual specialized learning and new collective knowledge that can be applied to the group's task*" (p.1259). TMS is enacted in tacit knowing of the location and awareness of team member expertise, which develops as team tacit knowledge is created.

This predictive model operationalizes the components presented in Figure 1. TMS relevance is defined as the strength of the relationship between a TMS and performance on a task [17]. But team performance is also a proxy for situational team knowledge since tacit knowledge is manifest in skilled team performance therefore it is thought that tacit knowledge plays a greater role than transactive memory in successful team performance.

Figure 2 Predictive Model for Acquisition and Sharing of Tacit Knowledge in Teams



Based upon the preceding discussion we advance two further hypotheses for the acquisition and sharing of tacit knowledge in software teams:

- ***H4**: Social interaction (quality and quantity) will predict team tacit knowledge above and beyond transactive memory.*
- ***H5**: Team tacit knowledge will predict team performance (efficiency and effectiveness) above and beyond social interaction (quality and quantity) and transactive memory.*

5 RESEARCH STUDY AND METHOD

The aim of this study was investigate the relationships between social interaction, transactive memory and team tacit knowledge. Furthermore the predictive model for the acquisition and sharing of tacit knowledge in teams is tested and the influence of team tacit knowledge on team performance in software development teams is explored.

5.1 Sample Characteristics

Forty eight teams from 46 small to medium organisations based in Ireland and the UK participated in an online interactive questionnaire consisting of 75% (N=121) males and 25% (N=60) females. These two countries were chosen, as Irish and UK software industries share several commonalities, both countries are dominated by international companies with indigenous companies emphasising vertical or niche market. The samples were drawn from two software business directories.

Most participants (47%) were in the 31-40 age range with an average of 11.64 years of experience (SD 4.97). Team size varied from 2 to 12+, with the mean team size being 4.86 and an average within team response rate of 81.86%. It was deemed that non-response bias was not a pervasive threat to the validity of the study, since, some non-participating organisations returned emails giving reasons for not taking part. These reasons included: lack of time, software development was not team based or they were not software developers. Given the challenges associated with getting teams rather than individuals to respond, the overall organisational response rate of 9.15% compares favourably to email studies and to studies involving industrial teams and software developers. The Irish sample was obtained from the National Software Directorate's (NSD) listing of over 700 Irish based software companies. The UK sample was obtained from the Kellysearch directory of over 3,000 UK software companies. In order to maximise response rates the following criteria were used to eliminate organisations from the sample frame:

If the target organisation

(a) did not provide an email address,
(b) did not develop software in teams,
(c) conducted software development 'offshore',
(d) were software re-sellers,
(e) were involved in software production maintenance,
(f) were involved in computer training, consulting or web design.

In all, 263, Irish, small to medium enterprises (SMEs) were contacted (29 emails were incorrect or returned by anti-spamming tool) yielding 234 usable contacts. The UK sample frame consisted of 382 SMEs (48 emails were incorrect or returned by anti-spamming tool). To further maximise response rates, where possible, the company CEO or COO names were obtained and all were contacted by email. The email explained the study and asked the recipient to forward it to the relevant project manager, who was asked to deploy the attached link to all team members. In addition, the email advised of anonymity of responses and offered a summary of the key findings of the study customised to each participating company.

5.2 Synopsis of Research Strategy

A survey design was chosen to measure the variables of interest in the present study. The survey method employs a number of instruments to collect data on all the variables of interest and provides a quantitative description of a sample population of software development teams through the use of self-report measures. Findings from the survey method may be generalised to the population of software development teams as a whole. In addition, the survey was developed for completion online.

An online survey was used because this distribution method best addressed the questions under study and suited the IT informed, time limited, participants. The survey developed in the present study, consists of a variety of both previously validated instruments and measures developed specifically for the present research. All measures, in the survey, involved self-report perceptions where participants quantified how often or how intensely they experienced the phenomena under study enabling comparisons across teams. The choice of survey items and the development of the tacit knowledge measure are detailed in the Section 5.3.

5.3 Measures and Scoring

The first section of the questionnaire detailed the study and ensured anonymity. Completion of the questionnaire took approximately 15 minutes and included a measure of Team Tacit Knowledge (TTKM) designed specifically for software development teams, an index of the quality of social interaction within the team, a measure of quantity of social interaction, a validated measure of transactive memory. In addition team performance was also assessed. The measures are now described.

Team tacit knowledge
Team tacit knowledge was measured using the Team Tacit Knowledge Measure (TTKM) [2]. The TTKM was scored by comparing the individual score on each of the 14 bipolar constructs items with an expert profile. We scored the responses to the TTKM by calculating a squared Euclidean distance of the individual from that of the expert mean. Internal consistency for the TTKM as measured by Cronbach's coefficient alpha a=.71 at the team level. Given that the obtained team level reliability

falls within the range for other situational judgement tests and for those reliabilities obtained on previous measures of tacit knowledge then we consider the internal consistency of the team level score to be acceptable.

These individual scores were then aggregated to form a team score, and the r_{wg} (agreement within group measure) for the TTKM scale was .96, indicating homogeneity and that aggregating members' scores to the team-level of analysis was statistically justified. Using Messick's [70] unified validity framework, it was found that the TTKM was a reasonably internally valid and reliable measure of tacit knowledge at the team-level Showing convergent validity with quality of social interaction, divergent validity with explicit job knowledge and predictive validity with team performance [2].

Quality of Social Interaction

The quality of social interaction was assessed by a self-report questionnaire regarding two perceived outcomes of social interactions across team members, resulting in an index of social interaction. This measure was adapted from Chiu et al. [71] in which participants were asked to recall the most recent instance where they spent more than 15 minutes alone interacting face-to-face with each member of the team. For each team member participants were asked to (a) indicate on a 3 point scale whether they had attained their goal in the interaction (where 0 = ' not applicable', 1 = 'no', 2 = 'to some extent' and 3 = 'yes'), and (b) indicate the degree of change in their relationship with the other person after the interaction, also on a 3 point scale where 1 = 'got worse', 2 = 'remained the same' and 3 = 'got better'. Chiu et al. [71] found that of a possible range of 1 to 9, the overall index of perceived interaction quality ranged from 2.33 to 9 (N= 95, mean = 5.80, SD 1.09) for college students. In the present study there were 4 categories of response for both questions where an extra category (1 = 'not applicable') was added. This category was included to allow for the presence of a team member with whom the respondent does not have informal interaction with another team member. This lack of informal interaction may be an indicator of a weak social relationship, thereforethe value for 'not applicable' was not '0' since this number has to be multiplied to form the interaction index, a '0' would have indicated a non-response. Hence the possible range was from 1 – 16. For each interaction the responses to these two questions were multiplied to form an interaction quality index for that social interaction. All of the interaction quality indexes were averaged to form an overall index of perceived interaction quality for each team. This measure was deemed adequate as it could be applied to different groups, allowed for the interaction quality for each team to be assessed and lent itself to an online interactive survey method. In order to compare to the original study which had a maximum score of 9 the Chiu et al. [71] mean score was transformed by dividing by 9 and multiplying by 16, the new mean was 10.31 and the new SD was 1.94.

In addition, convergent and discriminant validity were also established. In line with Chiu et al.'s [71] analysis for each interaction, the responses to the two questions were multiplied to form an interaction quality index for that social interaction. All of the interaction quality indexes were averaged to form an overall index of perceived interaction quality for each individual. These scores were then aggregated to form a team score of quality of social interaction.

Quantity of social interaction

Quantity of social interaction was measured using the method by Levesque, et al. [72] in 62 student software development project teams. Each person rated how much they had worked with each other member of their team, using a 6 point scale that ranged from 0 = 'not at all' to 5 = 'a lot'. The total interaction score was calculated by dividing the actual amount of interaction by the total possible interactions with other members of the team. A team interaction score was calculated for each team by taking the mean of its members' interaction scores. For Levesque et al [72] the team interaction score ranged from 0.28 to 0.81, with a mean interaction score of 0.54, on a scale of 0 to 1.00. This method was used because it was valid for software development teams and was appropriate for use in an online survey.

Transactive Memory

Transactive memory was gauged using the 15 item field measure of transactive memory developed by Lewis [16] where the TMS is a latent, second order factor (transactive memory systems), indicated by three manifest, first-order factors (specialisation, credibility, coordination), each of which was indicated by five items.

This scale was used to measure transactive memory because it is a team level measure and the only field measure of this construct, and is deemed valid and reliable. Respondents were asked to 'think of the last project or milestone that this team completed' and then respond to each item on a scale of 1 to 5 where 1 = 'strongly disagree' and 5 = 'strongly agree'. In the original validation of the TMS instrument three studies were conducted: a laboratory sample of 124 teams ($\alpha = 0.86$, $\bar{x} = 54.16$, SD = 4.91), a field sample of 64 Master of Business Administration consulting teams ($\alpha = 0.92$, $\bar{x} = 55.59$, SD = 6.16) and a field sample of 27 teams from technology companies ($\alpha = 0.82$, $\bar{x} = 23.12$, SD = 1.98). The first two studies used structured equation modelling to establish reliability and validity however a weighted composite was used for study 3, because the sample size was too small to use structured equation modelling.

In relation to the TMS measure, a weighted composite score was computed as this was deemed suitable for the sample size. The TMS score was weighted by regressing the TMS factor on its sub-factors and items, while still taking into account the hypothesised measurement model. Scale weights are given by the regression coefficients. In this study the scale weights were as follows: specialisation: $R^2 = 0.53$, credibility: $R^2 = 0.79$, coordination:$R^2 = 0\ .67$. The scores for each sub-factor were multiplied by their scale weight the three were added together to make the weighted composite.

Team Performance

Two dimensions of performance for knowledge teams consisting of effectiveness and efficiency were measured. Objective measures of performance present difficulties in the IS field [63], since using objective measures assumes comparability across software projects or unique situations constraints, and this raises a new set of methodological measurement issues [73]. Using self-assessment of performance rather than stakeholder assessment was deemed appropriate in this context, for practical reasons (as just outlined), issues related to confidentiality and to maximise response rates.

The effectiveness measure constituted 5 items and asked how well teams performed, in relation to other software development teams they have known, on dimensions of work quality, team operations, ability to meet project goals, extent of meeting design objectives and reputation of work excellence. The efficiency measure had two items and dealt with adherence to schedule and budget. Responses for both effectiveness and efficiency were rated on a 1 to 5 likert-type scale from 'not very good' to 'excellent'

The effectiveness measure consisted of five items and asked how well teams performed, in relation to other software development teams they have known, on dimensions of work quality, team operations, ability to meet project goals, extent of meeting design objectives and reputation of work excellence. The efficiency measure had two items and dealt with adherence to schedule and budget. Responses for both effectiveness and efficiency were rated on a 1 to 5 scale from 'not very good' to 'excellent'. The r_{wg} for the effectiveness scale was .90 and .76 for efficiency. These measures were developed by Faraj and Sproull [50] who found reliabilities for stakeholder ratings of software development teams for the effectiveness measure of $\alpha = 0.86$ (mean = 4.07; SD = 0.48) and for the efficiency measure of $\alpha = 0.74$ (mean = 3.85; SD = 0.77).

6 RESULTS

First individual scores were calculated for all variables, which were then averaged for team-level analysis. Table 1 presents the means, standard deviations, and bivariate correlations among the study variables and the internal consistency of the measures.

Table 1 *Means, standard deviations, reliabilities and inter-correlations among study variables (N = 48)*

Variable	M	SD	1	2	3	4	5	6
1. TMS	41.94	4.38	(.88)					
2. QSI	12.83	1.88	.61**	(NA)				
3. Quant SI	64.60	17.06	.64**	.39**	(NA)			
4. TTKM	5.49	2.48	.30*	.45**	.17	(.71)		
5. Effectiveness	3.69	0.55	.35**	.13	.15	.35**	(.88)	
6. Efficiency	3.24	0.73	.18	-.03	-.08	.09	.56**	(.83)

*NB. Values in parentheses represent internal consistency reliabilities (Cronbach's alphas) obtained in this study *p<.05, **p<.01*

6.1 Hypothesised Relationships

Development of a TMS was positively and significantly related to both quality and quantity of social (r=.61 and r=.64, p<.01, respectively) providing support for H1. A significant positive relationship was found between quality of social interaction and team tacit knowledge (r=.45, p<.01), but the relationship between quantity of social interaction and team tacit knowledge, while positive in direction, was not significant (r=.17, p>.05). This indicates that the nature and quality of the informal social interactions are related to the amount of tacit knowledge within a team, while the frequency of interaction is not, providing partial support for H2. On the other hand both quality and quantity of social interaction are important for the development of a TMS. A significant positive relationship was found between TMS and TTKM (r=.30, p<.05) as predicted in H3.

6.2 Other relationships

Effectiveness was also correlated significantly with team tacit knowledge and TMS (r=.35 and r=.35, p<.01, respectively). A significant positive relationship was found between quality and quantity of social interaction (r=.39, p<.01) but neither was significantly related to effectiveness.

6.3 Predictive Model for the Acquisition and Sharing of Tacit Knowledge in Software Teams

H4 posited that social interaction (quality and quantity) would predict team tacit knowledge above and beyond transactive memory. To test this hypothesis a hierarchical regression was conducted to ascertain the extent to which quality and quantity of social interaction in software development teams accounts for unique variance in team tacit knowledge ratings (Table 2).

***Table 2**. Hierarchical regression for TMS and Social Interaction in predicting tacit knowledge (N = 48)*

	Step 1		Step 2						
Variables	β	t	β	t	Df	R^2	F	ΔR^2	ΔF
Step 1: Control Variables									
TMS weighted composite	.30	2.11*	.06	.31	1, 46	.09	4.44*		
Step 2: Social Interaction									
Quality SI			.43	2.52*					
Quant SI			-.04	-.23	3, 44	.20	3.76*	.12	3.20*

* p<.05

It was found that the overall model was significant accounting for 20% of the variance in team tacit knowledge. Quality and quantity of social interaction significantly describe 12% of variance in team tacit knowledge above and beyond transactive memory. However, transactive memory is also a significant predictor of team tacit knowledge (9% of variance).

6.4 Predicting Team Performance

It was expected (H5) that team tacit knowledge will predict team performance (efficiency and effectiveness) above and beyond quality of social interaction, quantity of social interaction and transactive memory (weighted composite score). However as efficiency was only related to effectiveness and not to any of the predictor variables, it was decided to conduct just one hierarchical regression with effectiveness as the dependent variable. The results are presented in table 3.

6.5 Predicting Effectiveness

Table 3 presents the results of the hierarchical regression analysis to ascertain the extent to which team tacit knowledge in software development teams accounts for unique variance in effectiveness ratings.

Table 3. *Hierarchical Regression of Effectiveness on Transactive Memory, Social Interaction and on Team Tacit Knowledge*

	Step 1		Step 2						
Variables	β	t	β	t	Df	R^2	F	ΔR^2	ΔF
Step 1: Control Variables									
TMS weighted	.52	2.46*	.50	2.46*					
Quality SI	-.13	- .73	- .27	-1.53					
Quant SI	- 14	- .75	- .12	- .70	3,44	.15	2.50		
Step 2: Team Tacit Knowledge									
TTKM			.34	2.29*	4,43	.24	3.37*	.10	5.25*

*p<.05, **p<.01

The results in Table 3 around 24 % of the variance in effectiveness is accounted for by all of the variables combined in the full model, which is statistically significant. Team tacit knowledge describes 10% of variance in effectiveness above and beyond transactive memory and quality and quantity of social interaction and is statistically significant.
However transactive memory alone also exerts an influence of effectiveness. This suggests that transactive memory may even be a more important factor than team tacit knowledge in predicting effectiveness in software development teams. Quality and quantity of social interaction acted as suppressor variables.

7 DISCUSSION AND CONCLUSIONS

Support for the theoretical model (Figure 1) for the 'Acquisition and Sharing of Tacit Knowledge in Teams' was demonstrated in the significant positive relationships between TMS, TTKM and Quality of social interaction. Transactive memory was found to be related to both the quality and quantity of social interaction, which is supported by previous research into related team mental model and expertise coordination (e.g. [50, 51]). Furthermore Team tacit knowledge was related to transactive memory providing evidence that tacit knowledge is created through social interaction, which increases the development of shared mental models. Support for this position comes from related studies into expertise coordination, where teams were found to coordinate their expertise implicitly and mutual knowledge was tacit [50, 51].

In addition to the general support found for theoretical model, a predictive model (Figure 2) for the 'Acquisition and Sharing of Tacit Knowledge in Teams' was presented. This model was tested and provides answers to the two questions posed in section 1: *How do software development teams acquire and share tacit knowledge? What roles do tacit knowledge and transactive memory play in successful team performance?* We will now address with these two questions.

7.1 How do software development teams acquire and share tacit knowledge?

Transactive memory and quality of social interaction both contribute to team tacit knowledge, with quality of social interaction playing a more important role. Social interaction and transactive memory provide a reasonable model to explain the development of team tacit knowledge, with the quality of social interaction being the key.

It is concluded that tacit knowledge is acquired and shared directly through good quality social interactions and through the development of a TMS. TMSs are important for the acquisition and sharing of team tacit knowledge because they enact 'collective minds' of teams. Quality of social interaction is however a more important route through which teams can learn and share tacit knowledge than is transactive memory. The frequency of interaction indirectly aids the acquisition and sharing of tacit knowledge since it leads to better quality interactions and a more developed TMS. This study treated quality and quantity as separate entities, which provided a more in-depth analysis of the influence of social interaction. Both quantity and quality of social interaction enable the development of TMSs.

7.2 What roles do tacit knowledge and transactive memory play in successful team performance?

A central expectation in the present study is the predictive capacity of team tacit knowledge, social interaction and transactive memory for team performance as measured by effectiveness and efficiency. It is concluded that team tacit knowledge and transactive memory are both important factors in the prediction of effectiveness but not efficiency. Team tacit knowledge does predict effectiveness above and beyond quality and quantity of social interaction and transactive memory and indeed, transactive memory may also predict effectiveness above and beyond team tacit knowledge. Transactive memory is a factor in successful team performance and is enacted in tacit knowing of the location and awareness of team member expertise. Therefore software development teams with highly developed TMSs will have greater levels of team tacit knowledge than teams with less developed TMSs. Therefore, team tacit knowledge and transactive memory within teams are significant factors in effective performance for software development teams.

7.3 *Reliability and Validity*

Overall the measures compare well to results from previous studies. The teams in this study had good average quality of social interaction in line with the findings from Chiu et al. [71]. In addition, the quantity of social interaction mean score was above that of the original study by Levesque et al. [72]. The weighted TMS composite score was higher than the weighted composite in the Lewis [16] technical team study indicating that the software development teams in this study, on average had a slightly more developed TMS than that team.

The mean for the TTKM had quite a large standard deviation indicating quite a bit of variation in team responses, with some teams having much tacit knowledge and others relatively little. In terms of efficiency and effectiveness, the present study compares well to the results found by Faraj & Sproull [50].

7.4 Implications

A recurring theme in studies on software development and agile development in particular, is human and social factors and how these factors affect, and are affected by, agile principles [13]. Team work and team communications can be seen as a transversal component of particular importance, as to build software effectively, there is a need for tight coordination among the various efforts involved so that the work is completed and fits together [74]. The agile software development approach emphasizes that teams should be self-managed and committed to high levels of communication. There is a large body of existing theory in organisation studies, management science, and information systems that could be applied to the agile approach to better understand, explain, and predict its effects and efficacy [75]. Research into the agile approach to software development has matured in the past five years. Nevertheless, a number of open questions remain, and the relevance and implications of certain fundamental organisational concepts are still not fully understood in this context [76, 77]. One such concept is team communication and effective tacit knowledge transfer.

Agile methods suggest that most written documentation can be replaced by informal communications among team members internally and between team and the customers with a stronger emphasis on tacit knowledge rather than explicit knowledge. Much store is placed in face-to-face interaction but not much explanation of the cognitive processes involved in the creation of team and individual tacit knowledge. The theoretical model presented in Figure 1 provides a description for the way in which the agile principles of team tacit knowledge, social interaction and transactive memory relate to one another in the acquisition and sharing of tacit knowledge. The predictive mode presented in Figure 2 provides a testable model for the process while the findings from this study provide empirical support for these agile principles and their impact on effective performance.

This research highlights the importance of the TMS and team tacit knowledge in helping organisations differentiate between low- and high-performing teams in terms of effectiveness by suggesting that members of high-performing teams have developed a TMS and created tacit knowledge which is then applied to team tasks which can be seen in performance. This is useful in the agile approach as TMS may provide a proxy measure for success.

It is concluded that it may be more effective to have functionally rich teams in one location rather than distributing functions. Domain knowledge within the team skill-set needs to fully reflect and encompass all task components to complete the project/product. The best approach is teams in same place working on same task – it may be argued that you can geographically separate 'doers' from managers as in the case of some offshore teams with the caveat that there remains frequent and rich communication between the manager and at the very least the team leaders, e.g. daily scrum 'style' progress calls. In addition the effective use of issue tracking tools which are a central part of agile approach in practice.

Software development teams must have not only technical expertise but the relevant domain knowledge to understand the significance of the task and requirements. Domain knowledge is gained through experience and shared through social interactions. Longer tenured teams working in a particular domain therefore will have more developed TMSs and tacit knowledge about the task than merely technical teams [50].

7.5 Limitations and future research

This study has the limitations associated with most field research. The research design was non-experimental and used a self-report survey. Regardless of the sophistication of the statistical techniques, causal inferences must be treated with caution when using non-experimental designs. The survey measure was deemed to be a valid and reliable instrument for use in teams and for the purposes of the present study and was constructed to eliminate common-method variance by following the recommendations of Podsakoff et al. [78, 79].

Another source of bias may stem from the performance measure in the present study which was a self-report. Although perceptual data undoubtedly introduces limitations through the possibility of increased measurement error, research has found that there are no significant relationship differences between subjective and objective measures of perceived of performance [80, 81]. As stated above, causal inference should be treated with caution and perhaps further studies using objective measures will add validity to our findings.

A further limitation of this study is that there is no way of knowing if the teams collaborated or interacted with one another while completing the questionnaire. However, the existence of standard deviations across responses, on all measures in all teams provides some support that the teams did not collaborate.

Precautions were taken to ensure a representative sample of SMEs from Ireland and the UK. The response rate was in line with other surveys of the software industry, but still not a representative as one would like. In addition, care was taken when selecting a sample frame to include only those organisations that engaged in software development in teams. This led to a conservative sampling frame, where some companies whose web-sites did not indicate explicitly the nature or content of their activities were eliminated.

In this study we did not differentiate between agile and non-agile teams but focussed instead on software teams in general. It is recommended that future research distinguishes more agile teams from less agile teams. In addition, further research into geographical distribution of teams and the impact on TMS development, tacit knowledge sharing and subsequent team performance is necessary to understand fully the best team configuration for successful software teams. Finally many other human factors are at play in the development of team tacit knowledge such as trust, leadership etc. The influence of these human factors may be far reaching and further research from a human factors perspective will develop our understanding of software development teams.

REFERENCES

[1] A. Qumer, B Henderson-Sellers, A framework to support the evaluation, adoption and improvement of agile methods in practice, J. Syst. Software, 81 (11) (2008) 1899-1919.

[2] S. Ryan, R.V. O'Connor, Development of a team Measure for tacit knowledge in software development teams, J. Syst. Software, 82 (2009) 229-240.

[3] M. Polanyi, The Tacit Dimension, Routledge, London, 1966.

[4] R. Nelson, S. Winter, An Evolutionary Theory of Economic Change, Harvard University Press, Massachusetts, 1982.

[5] P. Busch, D. Richards, C.N.G. Dampney. The graphical interpretation of plausible tacit knowledge flows. Conferences in Research and Practice in Information Technology, 24 (2003) Australian Computer Society.

[6] R.J. Sternberg, G.B. Forsythe, J. Hedlund, J. Horvath, R.K. Wagner, W.M. Williams, S.A. Snook, E.L. Grigorenko, E. L., Practical intelligence in everyday life, Cambridge University Press, Cambridge, 2000

[7] I. Nonaka, H. Takeuchi, The Knowledge Creating Company. Oxford University Press, New York, 1995.

[8] T. D.Wilson, The nonsense of knowledge management. Information. Research, 8(1), (2002) Paper no. 144. Retrieved July 7, 2012, from http://InformationR.net/ir/8- 1/paper144.html.

[9] T. Chau, F. Maurer, G. Melnik, Knowledge sharing: Agile methods vs. Tayloristic methods, in: Proceedings of the IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 9-11 June, IEEE Computer Society, Linz, Austria, 2003.

[10] P. Clarke, R.V. O'Connor, The situational factors that affect the software development process: Towards a comprehensive reference framework, Inform. Software Tech, 54 (5) (2012) 433-447

[11] S. Nerur, V. Balijepally, Theoretical reflections on agile development methodologies, Commun. ACM, 50 (2007) 79–83.

[12] H. Sharp, H. Robinson, Three 'Cs' of agile practice, in: T. Dingsoyr et al. (Eds.), Agile Software Development: Current Research and Future Directions, Springer, Berlin, 2010, pp. 61–85.

[13] T. Dyba, T. Dingsoyr, Empirical studies of agile software development: A systematic review, Inform. Software Tech., 50 (9-10) (2008) 833-859.

[14] D.M. Wegner, Transactive memory: A contemporary analysis of the group mind, in: B. Mullen & G. R. Goethals (Eds.), Theories of Group Behavior, Springer-Verlag, New York, 1987, pp. 185-208.

[15] R.L. Moreland, Transactive memory: Learning who knows what in work groups and organizations, in: L. L. Thompson, J. M. Levine, & D. M. Messick (Eds.), Shared cognition in organizations: The management of knowledge, Erlbaum Mahwah, New Jersey, 1999, pp. 3-31.

[16] K. Lewis, Measuring transactive memory systems in the field: Scale development and validation, J. Appl. Soc. Psychol., 88(4) (2003) 587-604.

[17] K. Lewis, B. Herndon, Transactive memory systems: current issues and future research directions, Organ. Sci., 22(5) (2011) 1254-1265.

[18] R. Barker, M. Camarata, The role of communication in creating and maintaining a learning organization: Preconditions, indicators and disciplines. J. Bus. Comms., 35 (1998) 443-467.

[19] M. B. Pinto, J.K. Pinto, Project team communication and cross-functional cooperation in new program development. J. Product Innovation Management, 7 (1990) 200-212.

[20] T. Lechler, Social interaction: A determinant of entrepreneurial team venture success. Small Bus. Econ., 16 (2001) 263-278.

[21] A.B. Hollingshead, Communication, learning, and retrieval in transactive memory systems. J. Exp. Soc. Psychol., 34 (1998) 423-442.

[22] D.W. Liang, R. Moreland, L. Argote, Group versus individual training and group performance: The mediating role of transactive memory. Pers. Soc. Psychol. B., 21(4) (1995) 384-393.

[23] R.L. Moreland, L. Myaskovsky, Exploring the performance benefits of group training: Transactive memory or improved communication? Organ. Behav. Hum. Dec., 82(1) (2000) 117-133.

[24] A.E. Akgün, J. Byrne, H. Keskin, G. S. Lynn, S. Z. Imamoglu, Knowledge networks in new product development projects: A transactive memory perspective. Inform. Manage. 42(8) (2005) 1105–1120.

[25] Y. C. Yuan, J. Fulk, P.R. Monge, N. Contractor. Expertise directory development, shared task interdependence, and strength of communication network ties as multilevel predictors of expertise exchange in transactive memory work groups. Comm. Res. 37(1) (2010) 20–47.

[26] K.E. Weick, K.H. Roberts K.H., Collective mind in organizations: Heedful interrelating on flight decks. Admin. Sci. Quart., 38 (1993) 357-381.

[27] S. Fernie, S. Weller, S.D. Green, R. Newcombe, R., Knowledge sharing: context, confusion and controversy, Int. J. Project Manage., 21(3) (2003) 177-187.

[28] J.C. Spender, The dynamics of individual and organizational knowledge, in: C. Eden, J. C. Spender (Eds.), Managerial and Organizational Cognition, Sage, London, 1998, pp. 13-39.

[29] J.C. Spender, R.M. Grant, Knowledge and the firm: Overview, Strategic Manage. J., 17 (1996) 5-9.

[30] P. Thompson, C. Warhurst, G. Callaghan, Ignorant theory and knowledgeable workers: Interrogating the connections between knowledge, skills and services, J. Manage. Studies, 38(7) (2001) 923-942.

[31] P. Busch, D. Richards C.N.G. Dampney, C. N. G. The graphical interpretation of plausible tacit knowledge flows. Conferences in Research and Practice in Information Technology, 24 (2003) Australian Computer Society.

[32] A.C. Edmondson, A. B. Winslow, R.M.J. Bohmer, G.P. Pisano, G. P., Learning how and learning what: Effects of tacit and codified knowledge on performance improvement following technology adoption. Decision Sci., 34(2) (2003) 197-223.

[33] W.G. Dyer, Team building: issues and alternatives, second ed., Addison-Wesley, Massachusetts, 1987.

[34] J.R. Hackman, The design of work teams, in: J. Lorsch, (Ed.), Handbook of Organizational Behaviour, Prentice-Hall, New Jersey, 1987, pp. 315-342.

[35] P Berger, T. Luckmann, The Social Construction of Reality, Anchor, New York , 1967.

[36] E. Von Glaserfeld, Radical Constructivism: A Way of Knowing and Learning. Falmer Press, Washington, 1995.

[37] A.C. Edmondson, The local and variegated nature of learning in organizations: A group-level perspective, Organ. Sci., 13(2) (2002) 128-146.

[38] D.M. Rousseau, R.J. House, Meso-organizational behavior: Avoiding three fundamental biases, J. Organ. Behav., 1(1) (1994) 13–30.

[39] G. Salancik, J. Pfeffer, A social information processing approach to job attitudes and task design, Admin. Sci. Quart., 23 (1978) 224–253.

[40] J.R Hackman, Group influences on individuals in organizations, in: M. D. Dunnette , L. M. Hough, (Eds.), Handbook of Industrial and Organizational Psychology, vol. 3, second ed., Consulting Psychologists Press, Palo Alto, CA, 1992, pp. 199-267.

[41] J.A. Johannessen, J. Olaisen, B. Olsen, Mismanagement of tacit knowledge: the importance of tacit knowledge, the danger of information technology, and what to do about it. Int. J. of Inform. Manage., 21 (2001) 3-20.

[42] T. Davenport, L. Prusak, Working Knowledge. Harvard Business School Press, Boston, 1998.

[43] M. Hansen, N. Nohria, T. Tierney, What's your strategy for managing knowledge? Harvard Bus. Rev., 77(2) (1999) 106-116.

[44] K. Koskinen, P. Pihlanto, H. Vanharanta, Tacit knowledge acquisition and sharing in a project work context, Int. J. Project Manage., 21 (2003) 281-290.

[45] S. Tsuchiya, Improving knowledge creation ability through organizational learning, IIIA, Proceedings of International Symposium on the Management of Industrial and Corporate Knowledge, 93 (1993) 87-95.

[46] M.S. Granovetter, The strength of weak ties, Am. J. Sociol., 78(6) (1973) 1360–1380.

[47] H. Tsoukas, Do we Really Understand Tacit Knowledge? In: M. Easterby-Smith, M. Lyles (Eds.), The Blackwell Handbook of Organizational Learning and Knowledge Management, Massachusetts, 2003, pp. 410-427.

[48] S.L. Berman, J. Down, C.W.L. Hill, Tacit knowledge as a source of competitive advantage in the National Basketball Association, Acad. Manage. J., 45(1) (2002) 13-31.

[49] D. Leonard , S. Sensiper, The role of tacit knowledge in group innovation, Calif. Manage. Rev., 40(3) (1998) 112-132.

[50] S. Faraj, L. Sproull, Coordinating expertise in software development teams. Manage. Sci., 46 (12) (2000) 1554-1568.

[51] I. R. McChesney, S. Gallagher, Communication and co-ordination practices in software engineering projects, Inform. Software Tech., 46 (2004) 473-489.

[52] L. Layman, L. Williams, D. Damian, H. Bures, Essential communication practices for extreme programming in a global software development team, Inform. Software Tech., (489) (2006) 781–794.

[53] D Mishra, A Mishra, S Ostrovska, Impact of physical ambiance on communication, collaboration and coordination in agile software development: An empirical evaluation, Inform. Software Tech., 54 (10) (2012) 1067-1078.

[54] G.M. Olson, J.S. Olson, Distance matters, Human–Computer Interaction 15 (2000) 139–178.

[55] K. Crowston, J. Howinson, C. Masango, The role of face-to-face meetings in technology-supported self-organizing distributed teams, IEEE T. Prof. Commun. 50 (3) (2007) 185–203.

[56] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, et al., Agile Manifesto, 2001. Retrieved from http://agilemanifesto.org/

[57] G. Melnik, F. Maurer, Direct verbal communication as a catalyst of agile knowledge sharing, Agile Development Conference, 22-26 June, Salt Lake City, Utah, 2004.

[58] B. Bahli, E.S. A. Zeid, The role of knowledge creation in adopting extreme programming model: an empirical study, in: ITI 3rd International Conference on Information and Communications Technology: Enabling Technologies for the New Knowledge Society, 2005,pp. 75-87.

[59] P. Guinan, J. Cooprider, S. Faraj, Enabling software development team performance during requirements definition: A behavioral versus technical approach. Inform. Syst. Res., 9(2) (1998) 101-125.

[60] J.A. Cannon-Bowers, E. Salas, S.A. Converse, S. A. (1993). Shared mental models in expert team decision making, in: N. J. Castellan (Ed.), Individual and Group decision Making, Erlbaum, New Jersey, 1993, pp. 221-246.

[61] J. Orasanu, E. Salas, Team decision making in complex environments., in: G. A. Klein, J.Orasanu, R. Calderwood, C. E. Zsambok (Eds.), Decision Making in Action: Models and Methods Ablex, New Jersey, 1993, pp. 327-345.

[62] D.G. Ancona, D.F. Caldwell, Bridging the boundary: External activity and performance in organizational teams, Admin. Sci. Quart., 37 (1992) 634-665.

[63] J. Henderson, S. Lee, Managing I/S design teams: A control theories perspective. Manage. Sci., 38(6) (1992) 757-777.

[64] B.W. Boehm, Software Engineering economics, Prentice-Hall , New Jersey, 1981.

[65] R.L. Daft, Organization Theory and Design, tenth ed., South-Western, Ohio, 2009

[66] F. Brooks, The Mythical Man-Month: Essays on Software Engineering, Addison-Wesley, Massachusetts, 1995.

[67] R.L. Glass, Building Quality Software, Prentice-Hall, New Jersey, 1992.

[68] E. Lesser, L. Prusak, Communities of practice, social capital and organizational knowledge, in: E. L. Lesser, M. A. Fontaine, J. A. Slusher (Eds.), Knowledge and Communities, Butterworth-Heinemann Boston, 2000, pp. 123-131.

[69] J Shotter, H. Tsoukas, Theory as Therapy: Wittgensteinian reminders for reflective theorizing in Organization and Management Theory in: H. Tsoukas, R. Chia, (Eds.) Philosophy and Organization, Theory Research in the Sociology of Organisations, Vol. 32, 2011, pp.311-342.

[70] S. Messick, Validity of psychological assessment: Validation of inferences from persons' responses and performances as scientific inquiry into score meaning, Am. Psychol. 50 (1995) 741-750.

[71] C. Chiu, Y. Hong, W. Mischel, Y. Shoda, Discriminative facility in social competence: Conditional versus dispositional encoding and monitoring-blunting of information, Soc. Cognition, 1(1) (1995) 49-70.

[72] L.L. Levesque, J.M. Wilson, D.R. Wholey, Cognitive divergence and shared mental models in software development project teams, J. Organ. Behav. 22 (2001) 135–144.

[73] W. Gibbs, Software's chronic crisis, Sci. Am. September (1994) 86-95

[74] D. E. Strode, S. L. Huff, B. Hope, S. Link, Coordination in co-located agile software development projects, J. Syst. Software, 85(6) (2012) 1222-1238.

[75] R.E. Kraut, L.A. Streeter, Coordination is software development, Commun. ACM, 38 (1995) 69-81.

[76] P. Abrahamsson, K. Conboy, X. Wang, 'Lots done, more to do': the current state of agile systems development research, Eur J. Inform Syst., 18 (2009) 281–284

[77] P.J. Agerfalk, B. Fitzgerald, S.A. Slaughter, Flexible and distributed information systems development: state of the art and research challenges, Inform. Syst. Res., 20 (2009) 317– 328.

[78] P. M. Podsakoff, D.W. Organ, Self-reports in organizational research: Problems and prospects, J. Manage, 12 (1986) 69-82.

[79] P.M. Podsakoff, S.B. MacKenzie, J. Lee, N.P. Podsakoff, Common method biases in behavioural research: A critical review of the literature and recommended remedies, J. Appl. Psychol., 88(5) (2003) 879-903.

[80] W. H. Bommer, J.J. Johnson, G.A. Rich, P.M. Podsakoff, S.B. Mackenzie, On the interchangeability of objective and subjective measures of employee performance: A meta-analysis, Pers. Psychol. 48 (1995) 587-605.

[81] T. Wall, J. Michie, M. Patterson, S.J. Wood, M. Sheehan ,C.W. Clegg, M. West, On the validity of subjective measures of company performance, Pers. Psychol. 57 (2004) 95-118.