Tacit and Explicit Knowledge in Software Development Projects: A Combined Model for Analysis

Hanna Dreyer
The Business School
University of Gloucestershire
Cheltenham, UK
Dreyer.Hanna@gmail.com

Martin Wynn
School of Computing and Technology
University of Gloucestershire
Cheltenham, UK
MWynn@glos.ac.uk

***Abstract* – The development of new or updated software packages by software companies often involves the specification of new features and functionality required by customers, who may already be using a version of the software package. The on-going development and upgrade of such packages is the norm, and the effective management of knowledge in this process is critical to achieving successful enhancement of the package in line with customer expectations. Human interaction within the software development process becomes a key focus, and knowledge transfer an essential mechanism for delivering software to quality standards and within agreed timescales and budgetary constraints. This article focuses on the role and nature of knowledge within the context of software development, and puts forward a combined conceptual model to aid in the understanding of individual and group tacit knowledge in this business and operational environment.**

Keywords – software development; tacit knowledge; explicit knowledge; project management; knowledge management; conceptual model.

I. INTRODUCTION

Knowledge management, and more specifically the relationship between tacit and explicit knowledge, has been the focus of some recent research studies looking specifically at the software development process [1] [2]. Tacit knowledge is difficult to articulate but is, according to Polanyi [3], the root of all knowledge, which is then transformed into explicit, articulated knowledge. The process of tacit to explicit knowledge transformation is therefore a key component of software development projects. This article constructs a model that combines elements from other studies, showing how tacit knowledge is acquired and shared from both a group and an individual perspective. It thus provides a connection between existing theories of tacit and explicit knowledge in the workplace, and suggests a way in which teams can focus on this process for their mutual benefit.

McAfee [4] discussed the importance of interpretation within software projects and the dangers of misunderstandings arising from incorrect analysis. Such misconceptions can be explicit as well as tacit, but, generally speaking, in software development, the majority of knowledge is tacit. Ryan [5] states that "knowledge sharing is a key process in developing software products, and since expert knowledge is mostly tacit, the acquisition and sharing of tacit knowledge …. are significant in the software development process." When there are several parties involved in a project, with each being an expert in their field, the process and momentum of knowledge sharing and its acquisition for onward development is critical to project success. In addition, de Souza *et al.* [6] argue that the management of knowledge in a software development project is crucial for its capability to deal with the coordination and integration of several sources of knowledge, while often struggling with budgetary constraints and time pressures.

Individual and group knowledge are essential to a project. Individual knowledge within a group is the expertise one can share. Essentially, expert knowledge is mainly tacit, and needs to be shared explicitly within the group to positively influence project outcomes. Polanyi [3] has noted that "we can know more than we can tell," which makes it more difficult for experts to transfer their knowledge to other project actors. To comprehend the transfer of tacit knowledge within a project group, both individual and group knowledge need to be analysed and evaluated. The study of the main players, and the people they interact with, can identify the key knowledge bases within a group. In a software development project group, this will allow a better understanding and management of how information is shared and transferred.

This paper comprises seven sections. The relevant basic concepts that constitute the underpinning theoretical framework are discussed next, and two main research questions are stated. The research methodology is then outlined (Section III), and the main models relevant to this research are discussed and explained in Section IV. Section V then discusses how these models were applied and developed through field research and Section VI combines elements of these models into a framework for understanding the transfer of tacit knowledge from an individual and group perspective. Finally, the concluding section pulls together the main themes discussed in the paper and addresses the research questions.

II. THEORETICAL FRAMEWORK

Knowledge helps the understanding of how something works and is, at its core, a collection of meaningful data put into context. There are two strategies to manage knowledge within a company, codification – making company knowledge available through systemizing and storing information – and personalization – centrally storing sources of knowledge within a company, to help gain access to information from experts [7]. This article mainly focuses on personalisation, and on how knowledge is passed on from one source to the next. Knowledge does not have a physical form, but rather remains an intellectual good, which can be difficult to articulate and cannot be touched. Tacit knowledge - non-articulated knowledge - is the most difficult to grasp. According to Berger and Luckmann [8], knowledge is created through social, face-to-face interaction and a shared reality. It commences with individual, expert, tacit knowledge, which can then be made into explicit knowledge. Social interaction is one of the most flourishing environments for tacit knowledge transfer. Through a direct response from the conversation partner, information can be directly put into context by the receiver and processed in order to enrich their individual knowledge. This interplay in social interactions can build group tacit knowledge, making it easier to ensure a common knowledge base.

Advocating the conversion of tacit into explicit knowledge, Nonaka and Takeuchi [9] view tacit knowledge as the root of all knowledge. A person's knowledge base greatly influences the position an actor has within a group during a project. The effectiveness the actor possesses to transform their expert, tacit, knowledge into explicit knowledge determines how central the actor is within the group, and whether the group can work effectively and efficiently. Transferring human knowledge is one of the greatest challenges in today's society because of its inaccessibility.

Being able to transfer tacit knowledge is not a matter of course - how to best conceptualize and formalise tacit knowledge remains a debate amongst researchers. Tacit knowledge is personal knowledge, which is not articulated, but is directly related to one's performance. Swan *et al.* [10] argue that "if people working in a group don't already share knowledge, don't already have plenty of contact, don't already understand what insights and information will be useful to each other, information technology is not likely to create it." Communication within a software development project is thus crucial for its success. Assessing vocalized tacit knowledge remains a field which is yet to be fully explored. Nonaka and Takeuchi [9] conceptualize knowledge as being a continuous, self-transcending process, allowing individuals as well as groups to alter one's self into a new self, whose world view and knowledge has grown. Knowledge is information put into context, where the context is crucial to make a meaningful basis. "Without context, it is just information, not knowledge" [9]. Within a software development project, raw information does not aid project success; only when put in a meaningful context and evaluated can it do so.

In a corporate context, to achieve competitive advantage, a large knowledge base is often viewed as a key asset. The interplay between individual, group and organizational knowledge allows actors to develop a common understanding. However, according to Schultze and Leidner [11] knowledge can be a double edged sword, where not enough can lead to expensive mistakes and too much to unwanted accountability. By differentiating between tangible and intangible knowledge assets, one can appreciate that there is a myriad of possible scenarios for sharing and transferring knowledge. Emails, briefs, telephone calls or formal as well as informal meetings, all come with advantages and disadvantages relating to the communication, storage, utilization and transfer of the shared knowledge. For the analysis of the roles played by different actors within a group, social network analysis [12] can be used to further understand the relationships formed between several actors. Key actors are central to the understanding of the origin of new ideas or technologies used by a group [13]. Within a software development project, a new network or group is formed in order to achieve a pre-determined goal. The interplay between the different actors is therefore critical to understanding the knowledge flow throughout the project.

The Office of Government Commerce defines a project as "a temporary organization that is needed to produce a unique and pre-defined outcome or result, at a pre-specified time, using predetermined resources" [14]. The time restrictions normally associated with all projects limits the time to understand and analyse the explicit and tacit knowledge of the people involved. The clearly defined beginning and ending of a project challenges the transfer of knowledge and the freedom to further explore and evaluate information. Having several experts in each field within a project scatters the knowledge, and highlights the need for a space to exchange and build knowledge within the group. Software development project teams are a group of experts coming together in order to achieve a pre-determined goal. The skills of each group member must complement the others in order to achieve project success. Ryan [5] argues that group member familiarity, as well as the communication frequency and volume, are "task characteristics of interdependence, cooperative goal interdependence and support for innovation;" and that these are critical in software development groups in engendering the sharing of tacit knowledge. Faith and comfort in one another is essential to ensure group members transfer personal experience and knowledge with team mates. Tacit knowledge transfer in software development is central to the success of the project [15]. Researchers may argue about how effective the transfer of knowledge may be, but most agree on the importance and impact it has on project outcomes [16].

Communication issues are one of the key causes of project failure, where meaningful knowledge exchange is impaired. Furthermore, once a project is completed, the infrastructure built around a project is usually dismantled, and there is a risk that knowledge produced through it may be degraded or lost altogether. When completing a project, the effective storage and processing of lessons learned throughout the project, as well as the produced knowledge, can act as a platform for improved knowledge exchange and overall outcomes in subsequent projects. A significant amount of knowledge in software development projects is transferred through virtual channels such as e-mails, or virtual message boards, and the flow of knowledge has greatly changed in the recent past. Much of the produced knowledge is not articulated, which can lead to misconceptions and misunderstanding. This can be exacerbated in a software development environment, because of time limitations and the need for quick responses to change requests and software bug fixing.

In this context, this research seeks to answer the following research questions (RQs):
RQ1: How can tacit and explicit knowledge be recognised and evaluated in software development projects?
RQ2: Can tacit and explicit knowledge be better harnessed through the development of a combined model for use in software development projects?

III. Methodology

This research focuses on the identification of tacit knowledge exchange within a software development project, and aims to understand the interplay between individual and group tacit knowledge. A shared understanding between the main players and stakeholders is essential for a software development project as it is essentially a group activity [17]. Using a case study approach, the research is mainly inductive and exploratory, with a strong qualitative methodology. Validating the composition of several models, the aim is to understand the tacit knowledge flow in software development projects, and specifically in key meetings. Subjectivism will form the basis of the philosophical understanding, while interpretivism will be the epistemological base.

The aim is to show the topic in a new way, albeit building on existing models and concepts. Through data collection and analysis in a specific case study of software development, a model to understand the interplay between individual and group tacit knowledge is developed. The data is largely generated through unstructured interviews, and in project meetings, where the growth of knowledge has been recorded and assessed in great detail. This demands a narrative evaluation of the generated data and is therefore subject to interpretation of the researchers [18]. Participant observation and personal reflection also take part in forming and contextualizing the data.

As knowledge is qualitative at its core, textual analysis can also aid in the understanding and interpretation of meetings. Expert knowledge is sometimes worked on between group meetings, to be made explicit and exchanged within meetings. Current models can help in evaluating exchanged knowledge within meetings. As knowledge does not have a physical form, the information generated throughout the meetings needs to be evaluated in a textual form. The data generated from the meetings has helped develop an understanding of tacit knowledge within the software development project and its relationship to individual and group tacit knowledge. Different expert groups can have a major influence in determining the flow of knowledge in a project.

The data was collected over a three month period and amounts to approximately 30 hours of meetings. The data collection was project based and focused on the key people involved in the project. In total, there were ten people working on the project (the "project team") - four core team members who were present at most of the meetings, two executives (one of which was the customer, the other the head of the HR consultancy company) and one programmer. These were the players who had most influence on the project, hence the focus of most of the data was on them. The meetings were "sit downs" - usually between project team members and two of the HR consultants and a software development consultant. During these meetings, programmers joined for certain periods, and there were conference calls with the client and the head of the HR consultancy firm.

The topics discussed in the meetings were evaluated and contextualized, in order to analyse the knowledge exchange throughout the meetings. The data was evaluated systematically, where first the meetings were transcribed, then ordered according to topic. They were then categorized according to the theories of Nonaka, Ryan and Clarke. The first round of categorization mainly focused on topics discussed during the meetings and whether there was evidence of tacit knowledge surfacing. The second-round assembled related topics and transcribed the conversations. During this process, evidence of constructive learning, group tacit knowledge, individual knowledge, tacit knowledge triggers, as well as decision making, was searched for. The transcribed meetings were then organized in relation to the previously found evidence (constructive learning, group tacit knowledge etc.). Within this categorization, the meetings were still also classified by topic. Finally, during the last round of data evaluation, recall decisions and various triggers (visual, conversational, recall, constructive learning and anticipation) were searched for and identified.

Data analysis has supported the construction and testing of a model representing individual and group tacit knowledge. Personal reflection and constant validation of the data aim at eliminating bias in the interpretation of results.

In summary, the main elements of the research method and design are:

1. Qualitative exploratory research
2. Inductive research
3. Participant observation
4. Personal reflection
5. Unstructured interviews

This approach assumes that it is feasible and sensible to cumulate findings and generalize results to create new knowledge. The data collected is based on one project where knowledge passed from one group member to the other has been evaluated. The concepts of tacit and explicit knowledge are analyzed in a primary research case study. A key assumption is that there is a "trigger" that acts as a catalyst for the recall and transfer of different knowledge elements, and this is examined in the software development project context. These triggers are then related to previous findings in the data. The exchange of tacit knowledge over time, from a qualitative perspective within one project, allows the analysis of group members using previously gained knowledge from one another and its usage within the group.

IV. Relevant Models in Existing Literature

This research focuses on knowledge exchange in software development and aims to help future researchers analyse the impact of knowledge on project outcomes. It attempts to shed light on how knowledge builds within a group which can aid project success. This is done by creating a framework to represent the knowledge flow, from both an individual and a group perspective; but its foundations are found in existing theory and related models, and this section provides an overview of these.

Companies share space and generally reinforce relationships between co-workers, which is the foundation for knowledge creation. These relationships are formed in different scenarios throughout the work day. Some of the knowledge is formed through informal channels, such as a discussion during a coffee break, or more formally through e-mails or meetings. When such exchanges occur, whether knowledge be explicit or tacit, the "Ba" concept developed by Nonaka and Teece [19] provides a useful basis for analysis.

"Ba" is conceived of as a fluid continuum where constant change and transformation results in new levels of knowledge. Although it is not tangible, its self-transcending nature allows knowledge evolution on a tacit level. Through social interaction, knowledge is externalized and can be processed by the actors involved. It is not a set of facts and figures, but rather a mental ongoing dynamic process between actors, allied to their capability to transfer knowledge in a meaningful manner. "Ba" is the space for constructive learning, transferred through mentoring, modeling and experimental inputs, which spark and build knowledge. The creation of knowledge is not a definitive end result, but more an ongoing process. Nonaka and Teece [19] differentiate between four different elements of "Ba" - originating, dialoging, systemizing and exercising.

Individual and face-to-face interactions are the basis of *originating* "Ba". Experience, emotions, feelings, and mental models are shared, hence the full range of physical senses and psycho-emotional reactions are in evidence. These include care, love, trust, and commitment, allowing tacit knowledge to be shared in the context of socialization. *Dialoguing* "Ba" concerns our collective and face-to-face interactions, which enable mental models and skills to be communicated to others. This produces articulated concepts, which can then be used by the receiver to self-reflect. A mix of specific knowledge and capability to manage the received knowledge is essential to consciously construct, rather than to originate, new knowledge. Collective and virtual interactions are often found in *systemising* "Ba". Tools and infrastructure, such as online networks, groupware, documentation and databanks, offer a visual and/or written context for the combination of existing explicit knowledge, whereby knowledge can easily be transmitted to a large number of people.

Finally, *exercising* "Ba" allows individual and virtual interaction, which is often communicated through virtual media, written manuals or simulation programs. Nonaka and Teece [19] contrast *exercising* and *dialoguing* "Ba" thus: "exercising 'Ba' synthesizes the transcendence and reflection that come in action, while dialoguing 'Ba' achieves this via thought."

The ongoing, spiraling, process of "Ba" gives co-workers the ability to comprehend and combine knowledge in order to complete the task at hand. Establishing "Ba" as the basis of a combined model provides a secure framework anchored in existing theory, within which knowledge can be classified and understood. From the "Ba" model of knowledge creation, Nonaka and Teece [19] developed the SECI concepts to further understand the way knowledge moves across and is created by organizations. SECI – *Socialization, Externalization, Combination and Internalization* – are the four pillars of knowledge exchange within an organization (Figure 1). They represent a spiral of knowledge creation, which can be repeated infinitively, enabling knowledge to be expanded horizontally as well as vertically across an organization. This links back to the earlier discussion of tacit and explicit knowledge, as the four sections of the SECI model represent different types of knowledge transfer - tacit to tacit, tacit to explicit, explicit to tacit and explicit to explicit.

Socialization is the conversion of tacit to tacit knowledge through shared experiences, normally characterized by learning by doing, rather than consideration of theoretical concepts. *Externalization* is the process of converting tacit to explicit knowledge, where a person articulates knowledge and shares it with others, in order to create a basis for new knowledge in a group. *Combination* is the process of converting explicit knowledge sets into more complex explicit knowledge. *Internalization* is the process of integrating explicit knowledge to make it one's own tacit knowledge. It is the counter part of *socialization*, and this internal knowledge base in a person can set off a new spiral of knowledge, where tacit knowledge can be converted to explicit and combined with more complex knowledge.

The SECI model suggests that in a corporate environment, knowledge can spiral horizontally (across departments and the organization as a whole) as well as vertically (up and down management hierarchies). As we are focusing mainly on tacit knowledge, *combination* will not be part of the adopted model, due to its purely explicit knowledge focus. SECI helps us view the general movements of knowledge creation and exchange within companies.

Ryan's Theoretical Model for the Acquisition and Sharing of Tacit Knowledge in Teams (TMTKT) [5] [20] is also of relevance. Through a quantitative research approach, Ryan analyses the movement of knowledge within a group and the moment of its creation. Beginning with current team tacit knowledge, constructive learning enhances individual knowledge, which can then again be shared within the team in order to build up what Ryan terms the "transactive memory", which is a combination of specialization, credibility and coordination, resulting in a new amplified team tacit knowledge. This new team knowledge then begins again, in order to elevate the knowledge within the group in a never ending spiral of knowledge generation.

When developing the TMTKT, Ryan made several assumptions. First, team tacit knowledge would reflect domain specific practical knowledge, which differentiates experts from novices. Secondly,

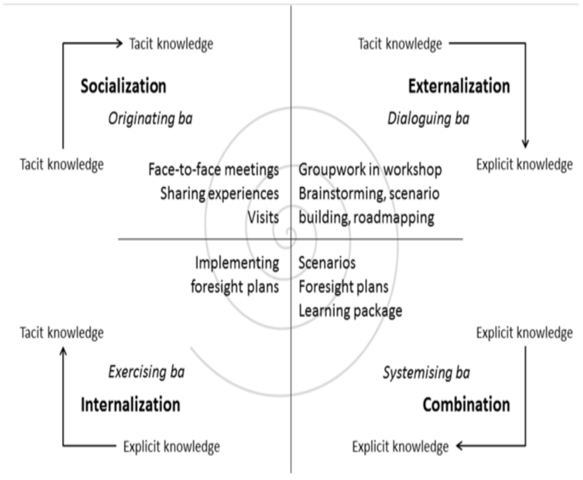

Figure 1. The Socialization, Externalization, Combination and Internalization (SECI) model [19]

the TMTKT needs to measure the tacit knowledge of the entire team, taking the weight of individual members into account. Finally, only tacit knowledge at the articulate level of abstraction can be taken into account. The model (Figure 2) comprises five main components or stages in the development of tacit knowledge:

1. Team tacit knowledge (existing)
2. Tacit knowledge is then acquired by individuals via constructive learning
3. This then becomes individual tacit knowledge
4. Tacit knowledge is then acquired through social interaction
5. Finally, the enactment of tacit knowledge into the the transactive memory takes place.

The starting point for understanding this process is to assess existing team tacit knowledge - this is their own individual tacit knowledge, but also includes any

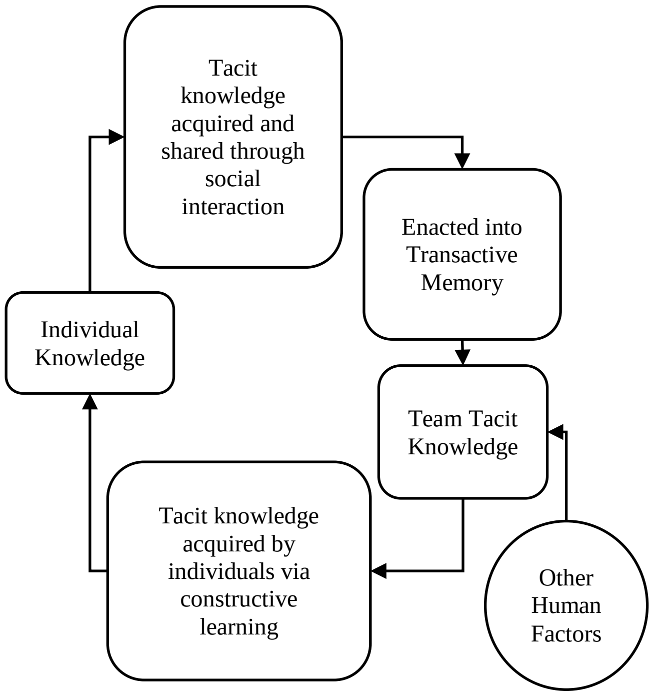
Tacit knowledge acquired and shared through social interaction
Enacted into Transactive Memory
Individual Knowledge
Team Tacit Knowledge
Tacit knowledge acquired by individuals via constructive learning
Other Human Factors

Figure 2. Theoretical Model for the Acquisition and Sharing of Tacit Knowledge in Teams (TMTKT) [5][20]

common understanding between the group members – that is, common tacit knowledge. Following group exchanges, new knowledge will be generated through constructive learning, building upon the original team tacit knowledge. The gained knowledge can then be made part of their own individual knowledge. These final two stages are a result of the social interaction, where team members gain knowledge and make it part of their transactive memory.

Clarke [21] evaluates knowledge from an individual point of view, and establishes a micro view of tacit knowledge creation. His model (Figure 3) suggests reflection on tacit knowledge can act as a trigger for the generation of new knowledge, both tacit and explicit. The process starts with the receiver being fed with knowledge – knowledge input - which is then processed, enhanced and formed into a knowledge output.

These models provide the basis for understanding the creation and general movement of knowledge in a software development project. Ryan and O'Connor [20] develops the idea of knowledge creation within a group further to specifically try to understand how knowledge is created and enhanced within teams. They provide an individual perspective of the flow of knowledge, which aids in the understanding how knowledge is processed within a person.

V. Testing and Validation Against Existing Models

During a three month period over 30 hours of meetings were recorded. The research mainly focuses on participant observation and the interaction between the project members. The conversations are analysed over this period, in order to see the development of learning over time; this aids in surfacing the range of acquired strategies which are applied in system development projects [22].

The project involved three different parties, who work on developing a cloud based human resource management software package. The developers of the software work in close contact with a human resource consultancy company, which is seeking a solution for their client. The meetings mainly consist of the developers and the human resource consultants working together to customise the software to suit the client's needs. No formal systems development methodology was used – the approach was akin to what is often termed "Rapid Application Development" based on prototyping solutions and amendments, and then acting upon user feedback to generate a new version for user review.

A total of ten actors were involved in the meetings, excluding the researcher. Each topic involved a core of six actors, where three executives took part in the decision making process, one from each company, and three employees, the head programmer and two human resource consultants. The software development executive acted in several roles during the project, performing as programmer, consultant and executive, this depending on the needs of the project.

This process involved the discussion of a range of topics, which encompassed payroll operations, recruitment, the design and content of software "pages" for the employees, a feedback option, absences, and a dashboard for the managers, as well as training for the employees. Throughout the meetings, changes were made to the software, these being at times superficial, such as choosing colour schemes, or more substantial, such as identifying internal processes where absence input did not function. The meetings relied on various channels for team communication, due to the client being in another city. Phone calls, face-to-face conversations as well as showing the software live through the internet were all used. These mediums were chosen in order to keep the client updated on the progress of the project, as well as giving input to their needs as a company. Once a week, a conference call was held with the three executives and their employees in order to discuss progress. The conference call helped

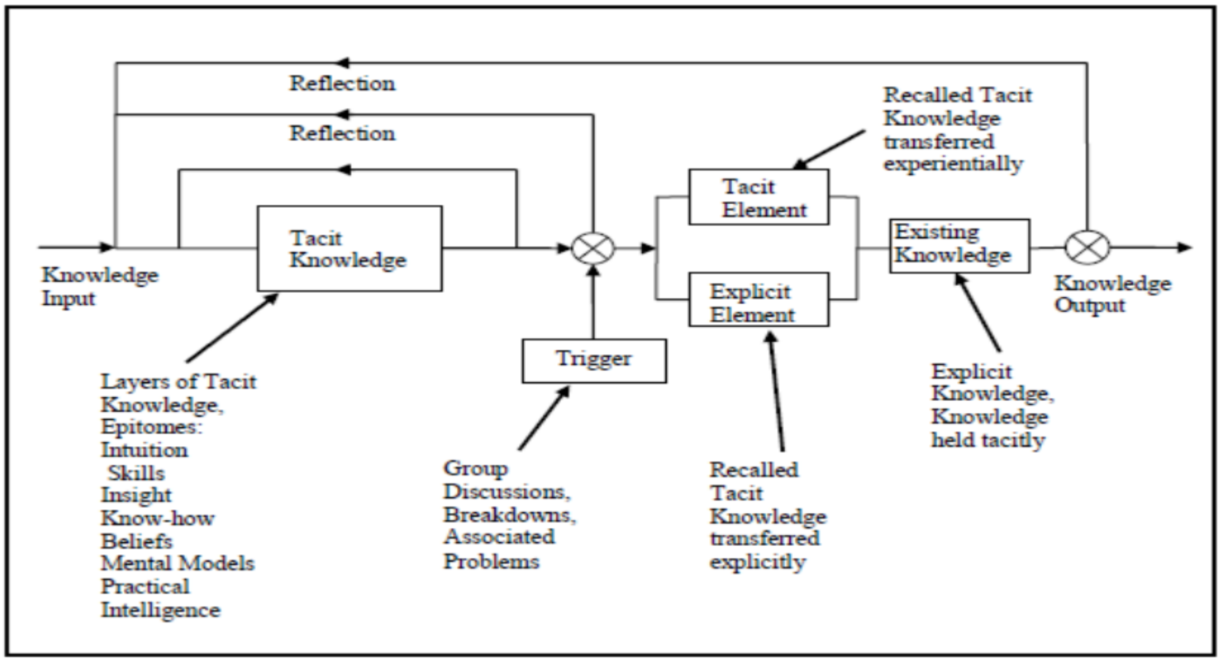
Reflection
Reflection
Recalled Tacit Knowledge transferred experientially
Tacit Element
Tacit Knowledge
Existing Knowledge
Knowledge Input
Explicit Element
Knowledge Output
Trigger
Layers of Tacit Knowledge, Epitomes: Intuition Skills Insight Know-how Beliefs Mental Models Practical Intelligence
Group Discussions, Breakdowns, Associated Problems
Recalled Tacit Knowledge transferred explicitly
Explicit Knowledge, Knowledge held tacitly

Figure 3. The Tacit Knowledge Spectrum [21]

ensure a common sharing of team tacit knowledge, allowing the different actors to then work on their individual tasks.

Knowledge regarding the different topics evolved throughout the project. The dynamic environment allowed different actors to request and exchange expert knowledge from the individuals. White and Parry [23] state that there has not been enough focus on the expert knowledge of developers, and how it affects the development of an information system. Expert knowledge from the developers and the interplay with the other teams supports White and Parry's findings. The data presented below illustrates conversations where expert knowledge is exchanged and utilized by team members. These exchanges helped validate the developed model, discussed below in Section VI.

One of the major issues that surfaced throughout the project was the complexity of the pension scheme of the end user. Integrating the correct values was vital for an accurate balance sheet and for payroll. The outsourced human resource management team of the user was not sure about certain aspects of the scheme, and needed the user human resource executive to explain in detail what was needed in order to make the software able to calculate a correct payroll. One of the outsourced HR consultants stated to the software development consultant: "Pensions is the most complicated thing. Ask the client on Monday to explain it to all of us." The HR consultant was thus suggesting the creation of a dynamic environment, where the software development company as well as the HR consultants themselves could learn about the pension scheme. On Monday, the consultant asked the client to explain their pension scheme: "I tried to explain pensions, but I could only do it poorly and I said that I only understand it when you (the client) explains it. So, could you please explain pensions to us, so that we are then hopefully all on the same page." The client went on to explain pensions, where occasional questions from the software developers as well as the HR consultants supported the comprehension of the group. By giving the client the opportunity to transfer his tacit knowledge, a "Ba" environment was created. This allowed group knowledge to emerge by the participants acquiring new knowledge and making it their own.

The analysis of these interactions provide the material for the construction of our combined model discussed below. We can see that knowledge input was given by the HR consultant through *socialization*, whereby tacit knowledge was shared through social interaction. This triggered the process of *internalization*, in which the user HR executive extracted tacit knowledge concerning pensions and transformed it into a knowledge output - *externalization*, being tacit knowledge acquired through constructive learning. This output was then received by each individual of the group, internalized and made into team tacit knowledge. At this point, the process starts anew, where unclear aspects are clarified by team members and externalized through social interaction. This process can lead the team to different areas of the discussed topic, where the input of different actors plays a vital role in shedding light on problems as well as identifying opportunities.

Another major issue that was in evidence during the project was the development of the time feature in the software. Within the time feature a calendar for sick leave, holidays or paternity was added. This was linked to the payroll since it was vital to know how much people get paid during which period. The interaction of the payroll and time in the software was very difficult to program. This part of the software was therefore explained by the programming executive, to make sure all needs of the customer were met. Here, the focus shifted to the HR consultants and the programmers. The end user was not involved in this process, since no expertise was needed and the user's main requirement was simply for it to work. The software consultant took a step aside, since this was not part of their expertise, and advised the software programmers to make sure the exchanged information was accurate. During the conversation not only were questions asked by the HR consultants, but also from the software development consultants. The programmer explained time and the time sheets, and during this discussion a knowledge exchange between the three parties created a dynamic environment, where group tacit knowledge was created.

Programmer: "We only want them to add days into the calendar where they should have been actually working – so that we can calculate the genuine days of holiday or leave. So if they are not due to work on a Monday, you don't want to count this as leave on a Monday. So it will only be inserted according to their working pattern."

Software consultant: "So the time sheet and calendar do the same thing?"

Programmer: "Yes, you choose against the service item, if the item should go into the calendar; so what will happen? - it will insert everything into the time sheet but then it will pick and choose which ones go into the calendar and which into the time sheet. So holidays will go into the calendar, but not go into the time sheet."

Software consultant: "You have a calendar in activities, which might show that a person is on holiday from x to y."

HR consultant: "But you might not want someone to know they are on maternity leave."

Software consultant: "But the time sheet is only working days, so you've got both options."

The conversation above demonstrates the process of knowledge input, *internalization*, output, group tacit knowledge as well as knowledge surfacing through a dynamic knowledge exchange. The programmer explains time sheets, which is then internalized by the software consultant, this then triggers a question, which leads to knowledge output. The programmer internalizes the question and creates a response through *socialization*. The spiral continues within the dynamic environment, and paves the way for knowledge to surface and to be used as well as internalized by the members of the team.

Throughout the analysis of the data this pattern of knowledge input, *internalization* and output was in evidence. This points to the significance of knowledge triggers to better understand the overall decision making process.

VI. Towards a Combined Conceptual Model

Building on the previous section, this section now examines how the theories of Nonaka, Ryan, and Clarke can be utilised in a new combined model which demonstrates how knowledge is created and built upon within a company at a group, as well as individual, level. The "Ba" concepts of Nonaka's SECI model provide the background framework that defines the dynamic space within which knowledge is created, although as noted above, the *combination* element is not used here as it deals exclusively with explicit knowledge (Figure 4). We will use the acronym SEI (rather than SECI) in the specific context of the combined model discussed in this section. The SEI concepts demonstrate the movement of knowledge, which can be continuously developed.

Ryan's TMTKT uses elements which overlap with Nonaka's SECI model. When combining the two an overlap in the processes can be found, although a more detailed view is provided by Ryan. When

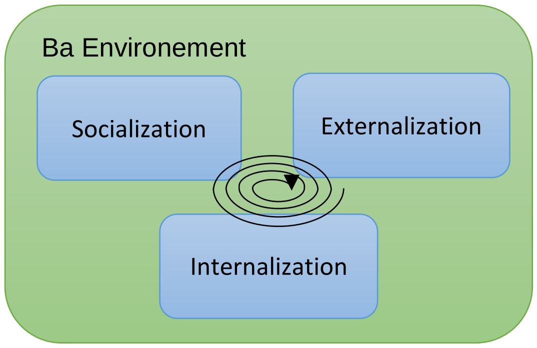

Figure 4. Three elements of SECI used in the conceptual model

analysing Nonaka's SECI, the process of internalization is explained in one step, unlike Ryan, who divides it into two steps rather than one. The *internalization* process is seen by Ryan as individual knowledge, which is then enacted into transactive memory, representing a deeper conceptualization of how people combine and internalize tacit knowledge.

According to the "Ba" concept, continuous knowledge creation is established within a dynamic environment, which supports the development of knowledge as it evolves from one stage to another. Figure 5 depicts how tacit knowledge is created, shared and internalized. Socialization indicates social interaction, where knowledge is acquired and shared through social interaction. Internalization entails making the knowledge one's own and combining it with previous knowledge, it being committed to transactive memory. Finally, externalization is knowledge acquired through constructive learning. Ryan's model focuses on tacit knowledge from a team perspective.

The last elements of the constructed model come from Clarke's Tacit Knowledge Spectrum [21]. This helps develop Nonaka's internalization process from a personal perspective. It provides a focus on one member of the team to complement Nonaka and Ryan's team perspective. Knowledge input commences the process, and different stages of knowledge intake make the knowledge individual knowledge. This focus on individual knowledge is encompassed in the internalization and enacted transactive memory stages of Nonaka's and Ryan's models, but it is treated in less detail.

Clarke's tacit knowledge spectrum commences with knowledge input, which is transformed into tacit knowledge. This tacit knowledge is then processed through reflection and at times, due to triggers such as additional information, the reflection process needs to be repeated in order to reveal new layers of tacit knowledge. The tacit and explicit elements permit additional layers of individual knowledge to be revealed, which can be through both explicit and tacit channels. Finally the new knowledge becomes part of the individual's existing knowledge. Existing knowledge can then once again be transferred into a knowledge output (Figure 3).

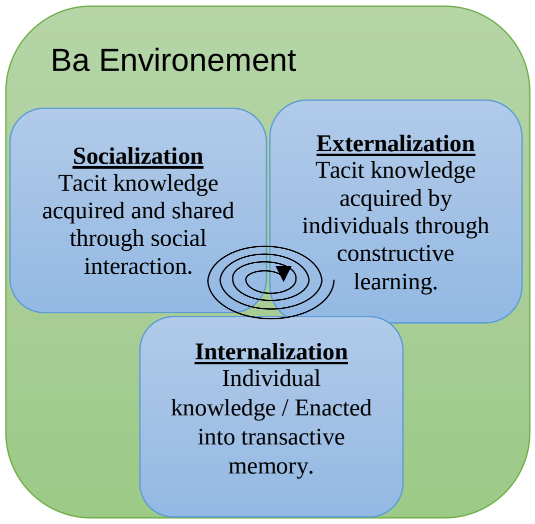
Ba Environement
Socialization
Tacit knowledge acquired and shared through social interaction.
Externalization
Tacit knowledge acquired by individuals through constructive learning.
Internalization
Individual knowledge / Enacted into transactive memory.

Figure 5. Combined elements of "Ba", SEI and TMTKT models.

Table I notes the main elements of the 3 approaches of Nonaka, Clarke and Ryan that are combined in the model used for case study analysis. At the macro level are Nonaka's concepts of "Ba", SEI and the spiral of knowledge. Ryan's model provides a group tacit knowledge perspective, complemented by Clarke's focus on the micro, individual knowledge generation process. The *internalization* and the *socialization* processes can involve both input and output, depending on the individual's point of knowledge acquisition – student or teacher.

TABLE I. ELEMENTS OF THE MODELS OF RYAN, CLARKE AND NONAKA USED IN THE COMBINED IGTKS MODEL

Nonaka	Ryan	Clarke
Socialization tacit to tacit	Tacit knowledge acquired and shared through social interaction.	Knowledge in- and output.
Externalization tacit to explicit	Tacit knowledge acquired by individuals through constructive learning.	Knowledge in- and output
Internalization explicit to tacit	Individual knowledge / Enacted into transactive memory.	Process of acquiring and processing tacit knowledge (reflection – trigger – tacit and/or explicit element – existing knowledge)

Nonaka's concept of "Ba" and its dynamic environment to support the exchange of knowledge provides the basis for a combined model, which we term the Individual and Group Tacit Knowledge Spiral (IGTKS). His theories also outline the different steps of the model, using *socialization* and *externalization* as knowledge in-and outputs, and the *internalization* process which represents individual knowledge. Clarke's model provides a more detailed view of the *internalization* process, which has been simplified somewhat in the combined model, concentrating on the trigger points, the reflection process and the enhancement of existing knowledge. Finally, Ryan's team tacit knowledge creates a point of "common knowledge" between the team members.

The combined model (Figure 6) aims to represent the process of continuous knowledge creation and exchange in a software development project team. The *internalization* process is an edited version of

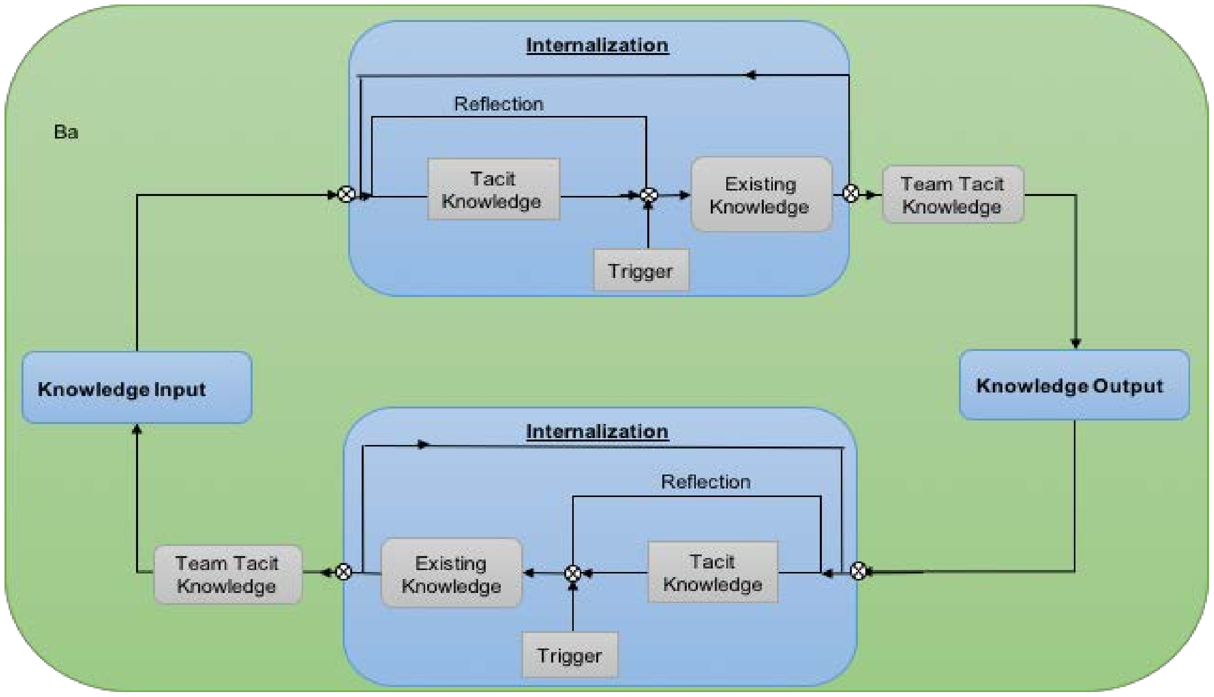

Figure 6. Individual and Group Tacit Knowledge Spiral (IGTKS)
(Triggers are represented by a circle with a cross in the middle)

Clarke's model - due to the focus on triggers it was not necessary to include the other elements of his model. The other human factors of Ryan's model were also modified, since the knowledge triggers entail the notion of personal experiences affecting tacit knowledge.

Knowledge is set in the "Ba" dynamic environment, where knowledge is freely exchanged and enhanced. The first step of the process is knowledge input, which can be knowledge exchange through social interaction or constructive learning. The knowledge input triggers the process of *internalization*.

Unlike Clarke, who only shows one trigger point, the IGTKS has three in every *internalization* process: one at the beginning, the initial trigger, which kicks off the *internalization* process; the second one is found after the development and combination of tacit knowledge which through reflection is developed to become a part of one's existing knowledge. The final trigger is at the end of the *internalization* process, where either the process is re-launched through an internal trigger or converted into team tacit knowledge. When the team arrives at the point where everyone has a common understanding of the knowledge, transferred through the initial knowledge input, then the team can react by sharing knowledge within the group via knowledge output transferred by *socialization* or constructive learning. This then again sets off the team members *internalization* process, where the knowledge put out by the team member is processed and embedded into their existing knowledge. Once this *internalization* process ends, the team has once again gained a common understanding of the exchanged knowledge and the cycle recommences. The data analysis demonstrated tacit knowledge creation and sharing through socialization, internalization and group tacit knowledge in 45 examples. Externalization was found 28 times, combination 9 and constructive learning 18 times (Figure 7).

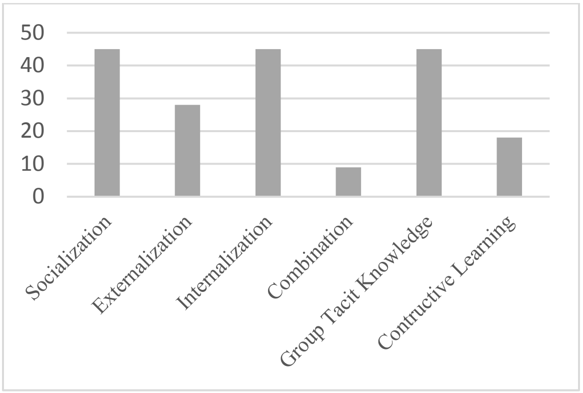

Figure 7. Number of examples of tacit knowledge sharing and creation in analysed conversations.

In addition, the trigger points showed conversations as the main factor in tacit knowledge acquisition and sharing, surfacing in 39 extracts. Visual triggers were shown to help tacit knowledge in 18 incidents, constructive learning 19, recall 7 and anticipation 2 (Figure 8).

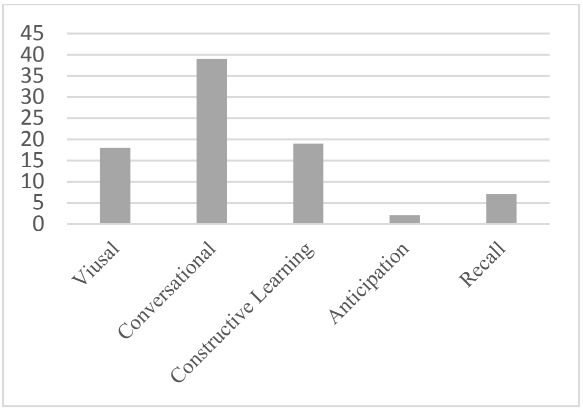

Figure 8. Number of tacit knowledge triggers identified in analysed conversations.

The IGTKS (Figure 6) can be used to model and analyze a conversation during a software development meeting, where, for example, team member A commences the meeting by asking a question about X. This question is then internalized by the other team members, B, C and D. A, B, C and D are now all aware that the topic of discussion is X, and understand the issue with it, and at this point the team has a common team tacit knowledge. However, topic X mainly concerns team member C, who therefore answers the question through knowledge output and constructive learning. Once C has explained X, the team again has a common team tacit knowledge. Now the cycle restarts, spirals, and other team members add knowledge within this dynamic knowledge environment.

Relating the model to this example, one can see how a conversation commences within the team. This then allows each individual to take the knowledge in, and make it their own tacit knowledge. During the internalization process, several triggers allow the creation of tacit knowledge. One of the triggers can be at the beginning of the internalization process - the unfiltered knowledge passed on by a project member which allows the internalization process to start. Then the knowledge is combined with previously gained knowledge; when newly received knowledge is complex, new thought processes can be triggered. Each individual then gains new tacit knowledge, which allows a new common group tacit knowledge. When the newly gained knowledge is incomplete, or when the receiver can complete or add to the knowledge, a response is triggered. This then commences the cycle to begin anew.

The aim of a meeting is to fill in gaps of knowledge within the project team, which allows teams to work together better. When the core people of a team or the expert within a field are not present, the project comes to a halt, until the knowledge is gained by the people in need of it. The model enables project teams to consider how knowledge is passed within the team. It demonstrates on a team, as well as on an individual level, the knowledge exchange process, and its limitations when key players are not present during a meeting. Utilizing knowledge from group members elevates the knowledge from each individual over time. Each member is needed to give input, and allow tacit knowledge to surface when needed. The process of absorbing knowledge, making it one's own tacit knowledge, and allowing a common base of group tacit knowledge to develop, can constitute a key influencer of project outcomes.

VII. CONCLUSION

Peter Drucker used to tell his students that when intelligent, moral, and rational people make decisions that appear inexplicable, it's because they see a reality different to the one seen by others [24]. This observation by one of the leading lights of modern management science underscores the importance of knowledge perception and knowledge development. With regard to software projects, McAfee [4] noted that "the coordination, managerial oversight and marshalling of resources needed to implement these systems make a change effort like no other". Yet, although software project successes and failures have been analysed within a range of analytical frameworks, few studies have focused on knowledge development.

Tacit knowledge in particular is one of the more complex and difficult aspects to analyse. Creating a well-functioning project team where knowledge can prosper within each individual is a great challenge, even more so when working within the time constraints of a software development project. Within this dynamic environment, tacit knowledge needs to flourish and evolve throughout the team, so each member can collect and harness information provided by the team to support task and overall project delivery. The comprehension of tacit knowledge processes within a software development project can help future projects enhance communication channels within the project to ensure project success. Project outcomes rely on the process of experts sharing their tacit knowledge, and building it up over the course of the project.

To return to the RQs noted in Section II, this research concludes that the combined theories of Ryan, Nonaka and Clarke can be used to establish an understanding of tacit knowledge, and provide a framework for recognizing it, in software development projects (RQ1). The analysis of meeting conversations provided the foundation for understanding the flow of knowledge within a software development project. Through the exploration of these conversations, different theories could be tested and applied, which helped to build the IGTKS model, demonstrating the knowledge interplay between different teams and people within the project. It facilitates the analysis of conversations on an individual as well as a group basis, to comprehend when an individual has received and processed information into knowledge. It seeks to demonstrate at what point the team has accepted information or knowledge as common group tacit knowledge, and in which circumstances more information or knowledge needs to be provided by other team members.

The combined model presented here can be used to further explore and evaluate knowledge flow on an individual and group level in software development projects. Unless it is rendered ineffective due to an absence of knowledge sharing, the knowledge spiral continues until common group tacit knowledge has been reached. The model allows the practitioner or researcher to pinpoint the moments where external and internal triggers launch the generation of tacit knowledge within an individual. This phenomenon requires further research into the interaction and communication of knowledge within and between project teams and their varying contexts, but this research suggests the combined model can be applied to better exploit tacit and explicit knowledge in the specific context of software development (RQ2). It supports the development of knowledge through a dynamic and open knowledge exchange environment, and suggests a way in which teams can focus on this for their mutual benefit. This can materially impact the software development process, and thus has the potential to significantly enhance the quality and subsequent functioning of the final software products.

REFERENCES

[1] H. Dreyer, M. Wynn, and R. Bown, “Tacit and Explicit Knowledge in a Software Development Project: Towards a Conceptual Framework for Analysis,” The Seventh International Conference on Information, Process and Knowledge Management, Lisbon, Feb 22nd – Feb 27th, ThinkMind, ISBN: 978-1-61208-386-5; ISSN: 2308-4375; 2015.

[2] F.O. Bjørnson and T. Dingsøyr, “Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used,” Information and Software Technology, Volume 50, Issue 11, October 2008, Pages 1055-1068, ISSN 0950-5849, http://dx.doi.org/10.1016/j.infsof.2008.03.006. (http://www.sciencedirect.com/science/article/pii/S0950584908000487

[3] M. Polanyi. The tacit dimension. The University of Chicago Press, 1966.

[4] A. McAfee, “When too much IT knowledge is a dangerous thing,” MIT Sloane Management Review, Winter, 2003, pp. 83-89.

[5] S. Ryan. Acquiring and sharing tacit knowledge in software development teams. The University of Dublin, 2012.

[6] K. de Souza, Y. Awazu and P. Baloh, “Managing Knowledge in Global Software Development Efforts: Issues and Practices,” IEEE Software, vol. 23, Issue 5, September, 2006, pp. 30 – 37.

[7] M.T. Hansen, N. Nohria and T. Tierney, “What is your strategy for managing knowledge?”, Harvard Business Review 77 (2), 1999, pp. 106–116.

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

[9] I. Nonaka and H. Takeuchi, The knowledge-creating company: How Japanese companies create the dynamics of innovation. Oxford: Oxford University Press, 1995.

[10] J. Swan, H. Scarbrough and J. Preston, “Knowledge management – the next fad to forget people?”, Proceedings of the Seventh European Conference on Information Systems, 1999, pp. 668–678.

[11] U. Schultze and D.E. Leidner, “Studying knowledge management in information systems research: discourses and theoretical assumptions,” MIS Quarterly, 26, 2002, pp. 213–242.

[12] C. Prell, Social Network Analysis: History, Theory and Methodology, 2012, London: Sage.

[13] T. W. Valente and R. Davies, “Accelerating the diffusion of innovations using opinion leaders,” The Annals of the American Academy of Political and Social Science, vol. 566, 1999, pp. 55-67.

[14] Office of Government Commerce (OGC), Managing Successful Projects with PRINCE2, 2009, London: The Stationery Office/Tso

[15] T. Clancy, The Standish Group Report Chaos, 1996.

[16] Z. Erden, G. von Krogh and I. Nonaka, The quality of group tacit knowledge, 2008.

[17] G. Fischer and J. Ostwald, “Knowledge management: problems, promises, realities, and challenges”, IEEE Intell. Syst., 16, 2001, pp. 60–72.

[18] T. Langford and W. Poteat, “Upon first sitting down to read Personal Knowledge: an introduction,” in Intellect and Hope: Essays in the thought of Michael Polanyi, 1968, pp. 3-18.

[19] I. Nonaka and D.Teece, Managing Industrial Knowledge, 2001, London: Sage.

[20] S. Ryan and R. O'Connor, “Acquiring and Sharing Tacit Knowledge in Software Development Teams: An Empirical Study,” Information and Software Technology, vol. 55, no. 9, 2013, pp. 1614 -1624.

[21] T. Clarke, The development of a tacit knowledge spectrum based on the interrelationships between tacit and explicit knowledge. 2010, Cardiff: UWIC.

[22] N. Vitalari and G. Dickson, “Problem solving for effective systems analysis: An experiential exploration,” Communications of the Association for Information Systems, 26(11), 1983, pp. 948–956.

[23] G. White and G. Parry, “Knowledge acquisition in information system development: a case study of system developers in an international bank,” Strategic Change, 25 (1), 2016, pp.81-95.

[24] B. Baker, “The fall of the firefly: An assessment of a failed project strategy,” Project Management Journal, 33 (3), 2002, pp. 53-57.