Tacit knowledge in the software development process: profile of a research case study

Hanna Dreyer

Address:
School of Computing and Engineering,
University of Gloucestershire,
Park Campus,
Cheltenham
Gloucestershire GL50 2RH
UK
Email:
dreyer.hanna@gmail.com

Martin Wynn
Address:
School of Computing and Engineering,
University of Gloucestershire,
Park Campus,
Cheltenham
Gloucestershire GL50 2RH
UK.
Email: MWynn@glos.ac.uk

Dr Hanna Dreyer was formerly on the staff of the International School of Management in Munich, and now manages a team of IT consultants in E-Learning. Previously, she obtained her BSc and MSc at Skema Business School focusing on Business and Project Management, and gained her PhD at the University of Gloucestershire, specializing in knowledge and IT management. Her research focus lies in tacit knowledge, IT management and project management.

Dr Martin Wynn is Associate Professor in the School of Computing and Engineering at the University of Gloucestershire. He obtained his BA and MA degrees at the University of Durham, and his PhD at Nottingham Trent University. He worked in industry as an IT professional for 20 years at Glaxo Pharmaceuticals and HP Bulmer, now part of the Heineken group. His research interests include digital transformation, technology transfer, and sustainability.

Abstract

This article reports on a case study of knowledge development in the customisation of a cloud-based software product. The aim of the research project was to understand the creation and explicit exchange of tacit knowledge in this environment, focusing on both the individuals and the project team. A phenomenological approach was adopted, based on participant observation, meetings recording and analysis, and grounded theory. Unlike most tacit knowledge research, the goal was to capture the creation of knowledge in its moment rather than by the participants' recall of the phenomena. Results highlighted the significance of five different types of trigger in new knowledge generation, turning tacit knowledge into explicit knowledge. The article concludes that, despite the recent evolution of the software development environment in the digital age, a qualitative, informal, people centric approach to analysing knowledge development and exchange is still an effective means to securing positive outcomes in software development.

Keywords: software development; exploratory research; ethnographic case study; grounded theory; tacit knowledge creation; project management; knowledge triggers.

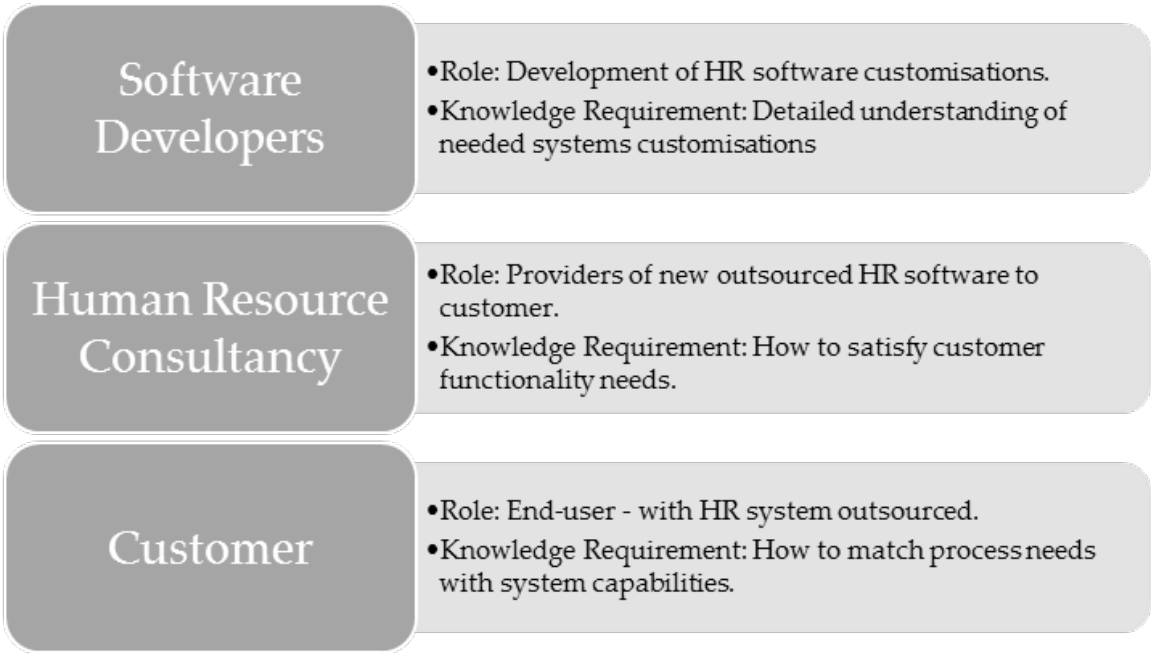
1 Introduction

This article discusses a research project that investigated the customisation of a cloud-based software product, through an in-depth exploratory investigation led by academics from the University of Gloucestershire, UK (Dreyer et al., 2020). The aim was to understand the creation and exchange of tacit knowledge in this environment, focusing on the individuals as well as the project team as a whole. The approach sought to clarify the moment of transformation from tacit to explicit knowledge, and to avoid individual interpretations of how and when this took place.

Data collection spanned a three-month period using participant observation, with a method of inquiry that was not focused on finding immediate answers. The day-to-day business involved in software development was therefore able to go on as usual, with the researcher being part of the work force. The project team consisted of three different parties, the customer, the software development company and a human resource (HR) consultancy. The aim of the customer was to update their payroll and HR system to a cloud based software product. The HR consultants were an outsourced company employed by the customer as their HR team. It was the aim of the HR company to receive a new tailored HR system for use with their customer. The software developers used the Oracle NetSuite® product as the basis for the new system, but customized and configured the product to the particular requirements of the customer. The cloud-based version of the software was used in order to allow a larger number

of people to access the system remotely. System customisations were developed by the three teams working in collaboration, with each of them exchanging and using their expert knowledge (Figure 1).

Figure 1 Roles and requirements of groups involved in the project



The platform to exchange this expert knowledge was a series of meetings, held three times a week, where expectations, changes and limitations of the software were discussed. These meetings were used as a basis for tacit knowledge exchange. The meaning and purpose of conversations were explored through a contextual methodology. Walsham (1995) suggests a broadly interpretive method to research the cues used to transform tacit into explicit knowledge and allow it to surface.

Tacit knowledge has been defined as the "articulable tacit", an individual and goal driven expert knowledge, which exits at the team level (Ryan and O'Connor, 2009). This tacit or expert knowledge is held by the members of a team, and allows each person to use, create and exchange the said knowledge for reaching a common goal. Knowledge is, according to Nonaka and Teece (2001), in a shared space wherein relationships emerge. In this context, the meetings within a software development project constituted the space. The overall goal was to reach software development project success. It is important to understand the constraints and opportunities such a project entails. Software projects are in many regards similar to other projects, time constrained and fast paced. It requires changes to the software to be done rapidly, and demands flexibility to meet evolving customers' needs. This provides for a fast expert knowledge exchange process – using an appropriate platform for knowledge exchange – with quick testing and implementation of successive software versions.

A detailed review of literature relating to tacit and explicit knowledge has been provided by the authors elsewhere (Dreyer et al., 2020), and this is not included here. Rather, the focus is on the theories underpinning this research process and method, which is detailed in section 2 of this article, and the research results, which are set out and discussed in section 3. Section 4 then reflects on the overall contribution of the research, its limitations and possible future work.

2 Research method and process

The purpose of an investigative study is to understand and research a specific phenomenon. Phenomenology aims to research the concerns of lived experiences by people (Maypole and Davies, 2001). It is the aim of such an interpretivist approach to inform others of a specific situation, unlike a positivist research approach, where the goal is to find a generalized perspective. Here, an interpretivist approach focuses on the words and experiences exchanged by the participants, as well as the researchers' perspectives. This is based on the epistemological position that social belief is directly related to the interpretation of meaning during a specific phenomenon, and to those interacting with it. This study was exploratory, analysing the situation with an inductive, phenomenological approach. Participant observation was used to understand how tacit knowledge was exchanged during the project. Tacit knowledge exchange - its moment of creation – was investigated through contextualizing the situation when, how and by whom, tacit knowledge is exchanged, created and internalized. The approach set out by Leonard et al. (2013) was followed for deep mentoring, allowing a form of

observing, practicing, partnering and joint problem solving (OPPTY), which helped to analyse the data.

The aim of a phenomenological approach is to understand and describe the lived experiences of people (Robinson and Reed, 1998). A phenomenological research approach is dictated by the phenomenon not by the method (Hycner, 1999). The participants' views, and the aim of understanding tacit knowledge in a software development project, influence the choices of methods used in the study (Welman and Kruger, 1999): a narrative approach is used to evaluate the intangible nature of tacit knowledge by recording the meetings between the different participants in the project. Vocal fluctuations can change through tonal cues (Wijetunge, 2012), which allow recordings to be more powerful than notes or transcripts. The research approach allowed a project to naturally emerge and grow by observing the interaction between participants.

Participant observation plays a vital role in phenomenology (Spradley, 1979). Anthropologist, for example, engage in participant observation in order to observe formal as well as informal interviewing (Agar, 1996). As a methodology, it entails the observer taking part in rituals, daily activities, interactions, and events of people being studied as one of the means of learning the explicit and tacit aspects of their culture (Dewalt et al., 2010). The method of participant observation supports the goal of understanding tacit knowledge at its point of creation, while embracing the cultural aspects of a software development project.

Data collection in the field played a vital role as observations may not be regarded as data unless a recording took place (Dewalt et al., 2010). Three main categories of field notes can be considered during research according to Seligman (1951), the first being recordings of observed events, through interviewing or conversations during events, the second are recordings of prolonged activities or ceremonies, where interviews cannot take place, and the third being a daily journal. This research focused on assessing a prolonged activity/ceremony - a weekly meeting for a three-month duration – aimed at delivering an acceptable and functioning human resource management software product. These were recorded as voice memos for further investigation after the fieldwork. Through maintaining a journal, feelings and events were recorded in relation to the research, as well as the events which took place. Not wanting participants to feel uncomfortable during the research (Jackson, 1990), only occasionally were notes taken during the meetings, or in front of the participants. An example of such an uncomfortable situation took place during one of the early meetings where a

participant remarked on the researcher taking notes "I am terribly worried about what the researcher is writing down." Although he was reassured, the taking of notes in meetings thereafter was strictly limited.

The research also draws upon grounded theory in the development of results. This theory sets out to understand the experiences of people in a rigorous and detailed manner. Concepts and categories, which emerge through textual analysis, are linked, and associated with substantive and formal theories. The results from grounded theory may rely on its intensity and the habitual process of analysing the data. According to Sadelowski (1995), the proofreading of the collected material is the beginning of the analysis since it starts to make "inchoate sense". Once the data collection is complete, the goal is to find a way to code it, to create themes or give a sense to the data. Ryan and Bernard (2000) argue that a researcher's aim is to "identify the range and salience of key items and concepts, discover relationships among these items and concepts, and to build and test models linking these concepts together". Through the transcript, a foundation for analysis was established, allowing the interviews to be processed into smaller samples of text. Then through listening, as well as a line by line analysis, a first basis was established for the subsequent identification of the five triggers discussed below. Proofreading of material is the groundwork of analysis, due to the material not making coherent sense at the beginning (Sadelowski, 1995).

Followed by a first proof reading, a code needs to be established to sort the data. Agar (1996) refers to this as "open coding" during grounded theory. This process allows the identification of emerging themes in the collected data. More and more categories and themes emerge through open coding, which can then be linked to theoretical models. Having several models of analysis in grounded theory allows different approaches to code the data. Strauss and Glazer (1967) define the "constant comparison method", where the focus lies on "when, why, and under what conditions, do these themes occur". Another is "memoing", where notes are taking to evaluate discovered concepts and practical matters, as well as to summarize the happenings in a research environment from the researcher's point of view. Strauss and Corbin (1990) later described these as code notes, theory notes and operational notes. Derived from the chosen model, grounded theorists frequently display their results by presenting segments of text, where quotes from participants are used to illustrate certain concepts and theories (Ryan and Bernard, 2000).

The findings from this research are based on a single case study. The case study is a widely used research method within business research, and such case studies can also be used for teaching and to support practice in a range of disciplines (Taylor and Wynn, 1982; Wynn et al., 2013). Bryman and Bell (2011) argue that the case study is particularly appropriate to be used in combination with a qualitative research method, allowing detailed and intensive research activity, usually in combination with an inductive approach as regards the relationship between theory and practice, as adopted here; and Saunders, Lewis and Thornhill (2018) argue that case studies are of particular value for explanatory or exploratory investigation, such as that pursued in this research.

The project was limited to thirty-four hours of recorded meetings, with the focus being on knowledge generation before the software was considered "live" (i.e., in operation). An in-depth examination of each conversation and the knowledge transferred allowed a thorough analysis of the meetings. A wide range of topics was discussed in these weekly meeting recordings (Table 1), in which the participants sat together and could exchange their expert knowledge amongst each another. The participants in the study were pre-determined by the project leaders. Participants needed to have had experiences relating to the phenomenon to be examined. This required expertise in human resource management, software engineering, consultancy and finance. Each participant added knowledge to the project dependent on their role and experience with the subject at hand.

Table 1 Main topics and duration of meetings recordings

Duration (hours: minutes)	**Main Topics Discussed**
02:38	Payroll and Sick Pay
00:45	Bureau Licensing
01:07	Payments
03:08	Time, Payroll
01:19	Recruitment
01:58	Recruitment
03:34	Payroll, Employee List, 360 reviews,

03:58	Licensing, Payroll, Project Plan
02:11	Payroll, Pensions, Reports, Dashboard
01:06	Searches
00:59	Time
02:24	Payroll & Data Entry
00:30	Training, Employee List
01:26	Time, Job Application
02:16	Payroll, Project Plan, Training
05.00	Various/Miscellaneous

The data collection process ended when the customized software product was launched, after the training sessions were held. Follow up interviews were not undertaken since the aim of the research was to capture knowledge creation and augmentation during the software development phase. The data collection ended with the new customized software going "live". Once the data from the recorded meetings and field notes was evaluated, it became clear that the field notes were not so much revealing the participants' knowledge exchange, but rather the personal thoughts of the researcher towards the different actors, and their role within the project. However, through the richness of the recorded meetings, the evaluation of knowledge being passed on from one participant to the other emerged, and their oral response became the main focus of the research. Unstructured interviews were part of these meetings, where the natural flow of conversation between project members permitted tacit knowledge to unfold and allowed its transformation to explicit knowledge. The collected data from the meetings was analysed in three cycles. During this period, themes and theories emerged. The inductive approach allowed theories from several researchers to be used and built upon in analyzing the data.

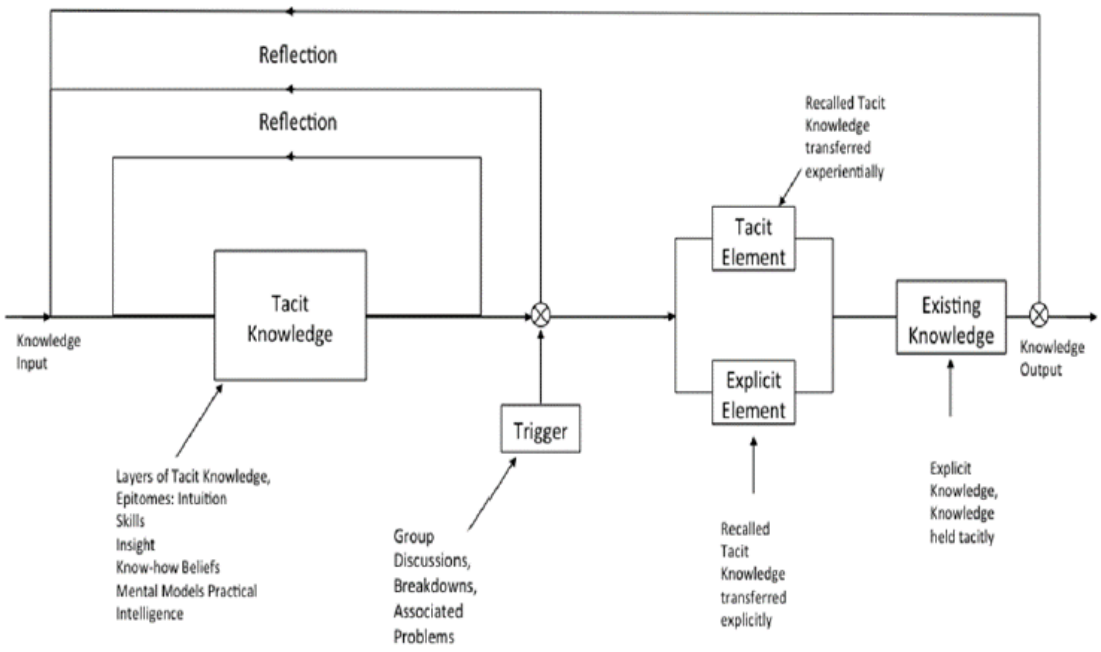
3 Findings

The relationship between tacit knowledge triggers and knowledge exchange was the most significant finding from the research. Triggers can be seen as unique events that start a process, initiating something new. The research suggested that triggers were the launch of turning tacit knowledge into explicit knowledge. The found triggers surface simultaneously at times, which enables knowledge to trigger through multiple channels. Building upon Clarke's Tacit

Knowledge Spectrum (Figure 2), five types of trigger emerged through the research. *Visual triggers* emerged by looking at previous notes or at the software, with tacit knowledge surfacing through this visual stimuli. In *conversational triggers*, tacit knowledge surfaces through a conversation held within the team. *Constructive learning triggers* were found regularly, when a team member explained a certain aspect of the software and tacit knowledge is enabled through the others learning. *Anticipation triggers* are related to an individual team member - tacit knowledge was exchanged by an individual in the group by waiting for the topic to come up or the meeting to take place. Finally, *recall triggers* lead to tacit knowledge resurfacing through discussions or visual aids, which had seemed forgotten. In all, eighty-three tacit knowledge triggers were identified in the thirty-four hours of meetings transcripts. Of these, 38 were conversational, 19 were constructive learning, 17 were visual, 7 were recall, and 2 were anticipation triggers. The topics discussed during the meetings were pre-determined by the initial specification by the customers of functional areas where amendments were required.

The triggers were then compared with moments of knowledge creation to further explore the point of tacit knowledge creation. This was done through the comparison of the triggers and their link to socialization, externalization, combination and internalization, and group tacit knowledge, as set out and defined by Nonaka and Takeuchi (1995). Conversational triggers allowed tacit knowledge to surface the most via socialization, internalization and group tacit knowledge development. Constructive learning triggers and visual triggers were the second and third triggers, which enabled tacit knowledge exchange. Here, tacit knowledge was mostly triggered through socialization, externalization, and group tacit knowledge development. Recall triggers and anticipation triggers were not found to be of such significance for tacit knowledge creation.

Figure 2 Clarke's tacit knowledge spectrum model



Source: Clarke (2010)

Given the context of this research, these findings suggest some points that are worthy of further discussion. First, software development is an environment where a quantitative research methodology dominates. Using a qualitative, inductive approach, however, can shed light on the human non-quantifiable aspects of the process. Software development project teams, which consist of members with a variety of disciplines and professional backgrounds, use their shared knowledge to create something new. Analysing this pool of knowledge allows a deeper understanding of how the software is produced, as well as how team members can aid one another to achieve project success.

Second, a people-centric approach to project management is likely to be most productive in generating new knowledge. Mubarak (2019), for example, notes that "with the recent advancement of technologies, it has been frequently observed that technology is dominating the discussions about knowledge management. Knowledge management comes in many different forms. One needs to understand that in any of these forms, technology does not manage knowledge but simply facilitates the interactions between people who possess or need knowledge". A people-centric approach is closely related to concepts of sustainable business models, in which organisations are in a constant state of evolution and change, within the context of their surrounding society and wider environment (Pedersen et al., 2018).

Third, this study involved knowledge sharing between three organisations, with a researcher embedded in the meetings process as observer. MacDonald (1994) suggested that formalizing such agreements could undermine informal networks and thereby hamper innovation and new knowledge creation. MacDonald's conclusions, however, pre-dated the internet and digital technologies era, which has seen a blurring of the boundaries between formal and informal interactions in business and the wider society. Nevertheless, it is still the case that attempts to structure and formalize the generation of new ideas and concepts may hamper the creation and sharing of new knowledge. In this research, an informal approach to participant observation, without pre-conceived ideas of what may surface, allowed the identification of triggers, which acted as the stimuli for new knowledge creation.

Fourth, there are implications for the relationship between collaboration within multi-functional teams and knowledge generation. Dance (2008) has suggested that collaboration within such teams can promote the association of ideas from different individual and disciplinary perspectives, and could also – through speeding feedback loops - facilitate the generation of new knowledge and ideas. Group collaboration in the development of a new software product, and the consequent generation and exchange of knowledge, will improve the likelihood of a fit-for-purpose product and successful project outcomes. The meetings allowed an assessment of explicit knowledge at its point of creation. This process was analyzed through the conversations that took place during the project. An example of this process is shown below in an extract from a meeting conversation, where the HR consultant, accountant and software developer discuss the import/upload of a payroll spreadsheet into the new system:

HR Consultant: We need to understand from your perspective [Accountant] the current process concerning import of data into the payroll system. Currently, when you receive the payroll data from the customer, you get the paper spreadsheet. Does that get imported electronically or is it manually keyed in?

Accountant: We manually key that in. It changes; it's not just static figures. It tells us what to do - we cannot ask the machine that. It is a very big spreadsheet. It tells us what the changes are and we manually have to change them, which is quite time consuming.

HR Consultant: We would like to work towards something that you do not have to key in, so it can be automatically imported. The Software Developer is with me and he is much more of an expert than I am. So what are your [Software Developer] questions in regards to what they need?

Software Developer: I think the situation that we've got is that we've got an interface (API) that we use in other payroll systems. This is how such data is automatically uploaded into the system. What would be really helpful is if we can keep to a standard format. It is quite similar to what you have now, except that we have several spreadsheets, rather than one. We have a spreadsheet with employee information such as their address, and other more sensitive information, such as salaries or bank details. Then we have another spreadsheet with their additions and deductions. So you would have their staff number appear there again. So if they had several allowances, clothing etc., they would have three separate entries on the spreadsheets. This gives the flexibility of adding as many additions as needs be. If a new allowance is created that various people get, in your current spreadsheet you would have that in columns, now they are created by types. Then there is a separate spreadsheet for addition of earnings and pension calculations.

The extract reveals the combination of tacit knowledge through a conversational trigger - explicit to explicit knowledge. Knowledge is combined and used to create a new common understanding of the subject. This example reveals several tacit knowledge traits – socialization, externalization, internalization, combination, group tacit knowledge and constructive learning. The import of the data and its processing is a tacit knowledge act where explicit knowledge is created. A conversational knowledge trigger enables the exchange of tacit knowledge by turning tacit into explicit, which then allows the other team members to internalize the new knowledge. Once internalized it is then processed and created into a new state of knowledge for each individual. With this new state, the internal knowledge can then be transformed into explicit knowledge, where the cycle begins anew.

Fifthly, the software development environment is evolving rapidly, not least because of advances in Artificial Intelligence (AI) and new opportunities for collaborative working offered

by cloud based development platforms. For example, recent research by Onyeka et al. (2020) reported on the development of an AI tool (COTIR) to integrate commonsense knowledge, ontology and text mining for early identification of implicit requirements in software development projects. The authors observe, "requirements engineering (RE) is a systematic process with many activities. A very important RE activity is requirements elicitation that harvests requirements (functional and non-functional) from stakeholders. Requirements can be classified into explicit and implicit. Explicit requirements are clearly stated and well-defined ones that a system should execute. Implicit Requirements (IMRs) are assumed or hidden requirements that a software system is expected to fulfill, though not directly captured during elicitation. The success or failure of a software development system highly depends on IMRs". This suggests that even though the mechanisms for tacit to explicit knowledge conversion may be changing as part of the overall digital transformation process, it remains a valuable part of the software development process.

4 Conclusion

This article illustrates the potential of another dimension to software development and team working. It is one case example, and provides insights into how knowledge is generated and developed, but is not a model from which wide-ranging generalisations should be made. Nevertheless, although the IT project manager may come armed with a plethora of methodologies and metrics, none of these addresses the issue of how knowledge is created and harnessed in support of achieving successful project outcomes. As noted by Sharp et al. (2016), "ethnographic studies help understand *how*, and more importantly, *why*, software teams do the things that they do." Knowledge and appreciation of this aspect of software development projects would be a worthy addition to the project manager's toolkit. Constructive learning within the project team and linked discussions to further understand software requirements can support successful project outcomes. Trigger points, which initiate the transfer of tacit knowledge into explicit knowledge, can only be created within a dynamic environment in which an exchange of knowledge is supported by the project team. Spending time together in a team environment is at the core of knowledge creation and transfer. Bouncing ideas off one another and subsequent mutual learning furthers the knowledge creation process. This allows each individual to take in more knowledge and provide a better, more complete view of the subject under discussion, enhancing the prospect of a more complete software product. Allowing the others to respond to explicit knowledge shared by a project member helps further

the knowledge within the group. A conversation is only then complete when the people involved have understood the needs and concerns of the other project members. As a project manager, a dynamic environment needs to be created where project members can exchange tacit knowledge and interact upon others. The aim of a meeting is to fill in gaps of knowledge within the project team, and allow teams to work together. In addition, creating triggers for team members to surface tacit knowledge can help the creation and sharing of tacit knowledge

There were nevertheless some limitations to this research and its applicability in practice. Research in this field is not easy because the topic is not tangible - the mind is being discussed and researched, and understanding and evaluating shared knowledge by each individual is challenging. A customizable, individualized, dynamic learning environment, where each project member can benefit from the knowledge of the group, is the ideal environment for tacit knowledge to surface. This can be created in the meetings of a qualified and resourceful project team. However, how do all the pieces fit together? Is it necessary to be in a conscious learning environment, or does it just come naturally? Are there certain tools that might facilitate transforming tacit knowledge into explicit knowledge? The more one studies the literature, analyses and evaluates cases, the more it is plain that no one seems to have a truly comprehensive or prescriptive template for what the perfect learning environment is. Turning what we cannot even articulate at times, into reproducible and effective explicit knowledge, and allowing others to memorize and internalize this knowledge, is not easy. That being said, building blocks and theoretical perspectives such as those developed here help analyse such situations, and allow us to have a glimpse of exchanged tacit and explicit knowledge within a group. Understanding the knowledge a team has is of increased importance in today's knowledge-based economy, where uncertainty and ambiguity can be all too common (Kolar, 2019).

The research was aimed to help practitioners pay more attention during projects to when and how tacit knowledge can be extracted from project members, and further investigations could productively focus on knowledge transfer with companies where, through a qualitative approach, tacit knowledge can be analysed. Understanding which factors allow tacit knowledge to surface, and its utilization during a project, can facilitate more rapid and productive outcomes. Other project dimensions may also be considered and their relevance to knowledge availability and transfer. For example, in a low skill software development team, explicit engineering knowledge might be a more fundamental need rather than tacit knowledge development. The necessity for, and value of, tacit knowledge might also depend on the process

maturity of the IT team and the organization for which they are developing software (CMMI Institute, 2020). Another dimension worthy of research is the impact of the cloud on the software development process itself, which, allied with advances in AI and robotics, may well see the type of interactions discussed here significantly modified, if not fully automated. Such research initiatives can shed new light on how to manage these issues in a rapidly evolving technology environment, to support mutual learning within the software development process, which will ultimately enhance the likelihood of project success.

References

Agar, M. (1996) *The Professional Stranger: An Informal Introduction to Ethnography*, 2nd ed., Academic Press, San Diego, USA.

Bryman, A. and Bell, E. (2011) *Business Research Methods*, 3rd ed., Oxford University Press, Oxford, UK.

Clarke, T. (2010) *the development of a tacit knowledge spectrum based on the interrelationships between tacit and explicit knowledge*. PhD Thesis, University of Cardiff, UK [online] https://repository.cardiffmet.ac.uk/bitstream/handle/10369/909/T%20Clarke.pdf (accessed 3 August 2020).

CMMI Institute (2020) *Build Capability, Drive Performance* [online] https://cmmiinstitute.com/ (accessed 8 October 2020).

Dance, J. (2008) *Five Reasons Why Collaboration Contributes to Innovation* [online] https://www.freshconsulting.com/5-reasons-why-collaboration-contributes-to innovation/ (accessed July 18 2020).

Dewalt, K.M., Dewalt B.R. and Wayland C.B. (2010) *Participant Observation,* 2nd ed., AltaMira Press, Lanham, USA.

Dreyer, H., Bown, G. and Wynn, M. (2020) 'Expert group knowledge triggers: when knowledge surfaces', *International Journal of Knowledge Management*, Vol. 16, No. 2, pp. 1-17 [online] http://eprints.glos.ac.uk/7150/ (accessed 27 July 2020).

Hycner, R. H. (1999) 'Some guidelines for the phenomenological analysis of interview data'. In Bryman, A. and Burgess, R. G. (Eds.), *Qualitative research,* Sage Publications, London, Vol. 3, pp. 143–164.

Jackson, J. (1990) 'I am a Fieldnote: Fieldnotes as a Symbol of Professional Identity'. In

Sanjek, R. (ed), *Fieldnotes: The Makings of Anthropology,* Cornell University Press, Ithaca, USA, p. 333.

Kolar, N. (2019), *Project Oxygen and Project Aristotle – What Makes Google, Google?* [online] https://medium.com/@nathankolar/project-oxygen-and-project-aristotle-what-makes-google-google-2a3fc79e5210

Leonard, D., Barton, G., Barton, M. (2013) 'Make Yourself an Expert', *Harvard Business Review* [online] https://hbr.org/2013/04/makeyourself-an-expert

MacDonald, S. (1994) 'Is Collaboration Good for Innovation?' *Industry and Higher Education,* Vol. 8, No. 3, pp. 141-146.

Maypole, J. and Davies, T. G. (2001) 'Students' perceptions of constructivist learning in a Community college American History II', *Community College Review*, Vol. 29, No. 2, pp. 54-80.

Mubarak, S. (2019), *Knowledge Economy World*. LinkedIn Post, Chairman, SPE Digital Energy Technical Section.

Nonaka, I. and Takeuchi, H. (1995) *The Knowledge Creating Company,* Oxford University Press, Oxford, UK.

Nonaka, I. and Teece, D. (2001) *Managing industrial knowledge: creation, transfer and utilisation*. Sage Publications, London, UK.

Onyeka, E., Vardey, A., Anuz, V., Tandon, N. and Daramola, O. (2020) 'Using Common sense Knowledge and Text Mining for Implicit Requirements Localization', *32nd International Conference on Tools with Artificial Intelligence (ICTAI*), IEEE, November, Chicago, Illinois. [online] https://www.researchgate.net/publication/346013371_Using_Commonsense_Knowledge_and_Text_Mining_for_Implicit_Requirements_Localization (accessed 22 November 2020).

Orlikowski, W. J. and Baroudi, J. J. (1991) Studying information technology in organizations: Research approaches and assumptions', *Information Systems Research*, Vol. 2, No. 1, pp. 1-28.

Pedersen, E., Gwozdz, W. and Hvass, K. (2018) 'Exploring the Relationship between Business Model Innovation, Corporate Sustainability, and Organisational Values within the Fashion Industry', *Journal of Business Ethics*, Vol. 149, No. 2, pp. 267-284.

Robinson, D. and Reed, V. (Eds.) (1998) *The A – Z of social research jargon,* Ashgate, Aldershot, UK.

Ryan, G.W. and Bernard, H.R. (2000) 'Data management and analysis methods', in Denzin, N. and Lincoln, Y. (Eds), *Handbook of Qualitative research*, 2nd Ed., Sage Publications, UK, pp. 769-802.

Ryan, S. and O'Connor, R. (2009) 'Development of a team measure for tacit knowledge in software development teams', *The Journal of Systems and Software*, Vol. 82, pp. 229-240.

Sandelowski, M. (1995) 'Qualitative analysis: What it is and how to begin' *Research in Nursing and Health,* Vol. 18, pp. 371-375.

Saunders, M., Lewis, P. and Thornhill, A. (2018) *Research Methods for Business Students*, 8th ed., Pearson Education Limited, England.

Seligman B.Z. (1951) *Notes and Queries on Anthropology*. 6th ed., Routledge & Kegan Paul, London, UK.

Sharp, H., Dittrich, Y. and de Souza, C. (2016) 'The Role of Ethnographic Studies in Empirical Software Engineering', *IEEE Transactions on Software Engineering* (Preprints).

Spradley, J. P. (1979) *The ethnographic interview,* Holt, Rinehart & Winston, New York, USA.

Strauss, A. L. and Corbin, J. (1990) *Basics of qualitative research: Grounded theory procedures and techniques,* Sage Publications, Newbury Park, California, USA.

Strauss, A. L. and Glazer, B. (1967) *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Aldine, Chicago, USA.

Taylor, J.L. and Wynn, M. (1982) 'Enhancing environmental education and training through case study experimentation', The Environmentalist, Vol. 4, Issue 3, pp. 235-242.

Welman, J. C., Kruger, S. J. (1999) *Research methodology for the business and administrative sciences,* International Thompson, Johannesburg, South Africa.

Wijetunge, P. (2012) 'Organizational storytelling as a method of tacit knowledge transfer: Case study from a Sri Lankan university', *International Information & Library Review,* Vol. 44, No. 4, pp. 212-223.

Walsham, G. (1995) 'Interpretive case studies in IS research: nature and method', *European Journal of Information Systems*, Vol. 44, pp. 74-81.

Wynn, M., Turner, P, and Lau, E. (2013) ‘E-business and process change in the UK SME sector’, *Journal of Small Business and Enterprise Development*, Vol. 20, Issue 4, pp. 913-933. [internet] http://eprints.glos.ac.uk/1382/ (accessed 17 April, 2021).