Master of Science in Software Engineering
January 2023

Tacit Knowledge Transfer in Agile Software Development

Gundlapally Preetham Raj Goud
Mechineni Prerana Rao

Faculty of Computing at Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Degree name. The thesis is equivalent to 20 weeks of full time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Author(s): Gundlapally Preetham Raj Goud
E-mail: prgu21@student.bth.se

Mechineni Prerana Rao
E-mail: prme21@student.bth.se

University advisor: KRZYSZTOF WNUK
Krzysztof.wnuk@bth.se
Department of Software Engineering Department

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

Background. Software companies make frequent development in their software products to improve their quality. Agile software development (ASD) helps in the rapid delivery of the software product with consistent quality to the customer. Agile Software development shares a lot of characteristics with knowledge-intensive works, and it also needs lot of knowledge from different domains of both human and computing domains. Knowledge is defined as a belief of one individual capability for an effective action. Tacit knowledge is a category of the knowledge management which is defined an individuals' memory, action or beliefs. Moreover, tacit knowledge plays a crucial role in ASD. However, tacit knowledge is hard to transfer among team members as tacit knowledge is based on an individual experience. Tacit knowledge is not documented for further reference so there is a need to study how efficiently can tacit knowledge is currently being transferred in the industry, what challenges are being faced in tacit knowledge transfer and the mitigation strategies used to overcome the tacit knowledge transfer challenges.

Objectives. In this present thesis, we focused on how tacit knowledge is being transferred among team members in agile software development.

Objective 1: To identify the current tacit knowledge transfer mechanism in agile software development.

Objective 2: To identify challenges in managing tacit knowledge transfer between team members in Agile software development.

Objective 3: To explore mitigation strategies to overcome the challenges faced during tacit knowledge transfer.

Methods. In this study, SLR and interviews were implemented to achieve the objective. SLR was used to achieve the first two objectives, and interviews were conducted to achieve all the objectives.

Results. From SLR, a total of 21 challenges and 12 transfer mechanisms have been identified whereas, from the interviews, a total of 12 challenges, 8 transfer mechanisms, and 7 mitigation strategies have been identified. There were new challenges and transfer mechanisms identified in both research methods. Some of the transfer mechanisms consist of daily scrum calls and day-to-day forums among the teams. Both the research methods' results indicate that one of the most challenging parts while transferring tacit knowledge is the lack of critical thinking with human orientation. Based on the interviews, some of the mitigation strategies such as regular sprint meetings, and online whiteboarding were considered to overcome the tacit knowledge transfer challenges.

Conclusions. Based on the challenges identified from the interview, it is evident that teams working in distributed teams are facing more challenges in transferring tacit knowledge, and tacit knowledge transfer sessions should be recorded to reduce the challenges.

Keywords: Tacit knowledge, Agile software development, challenges, mechanism.

Acknowledgments

Throughout the thesis period, we have got an amazing opportunity to study and gain knowledge about so many things. We take this chance to thank our supervisor Dr Krzysztof Wnuk for accepting to be our guide throughout this period and for being such a dedicated guide we could ever get. He has helped us every time we were facing a rockhead and was very patient to help us out whenever he was needed. Secondly, we would also like to thank our interviewees for taking time out of their busy lives and helping us out with the thesis with their responses. Without their responses, the thesis would never have come to a conclusion. I would also like to thank all the professors of Blekinge Institute of Technology for sharing their precious knowledge and experience with us.

“I personally have faced many challenges while working on our thesis due to a lack of knowledge in this field and lack of experience in conducting a thesis. I would like to thank my partner Prerana Mechineni for being so understanding and supportive and patient with me and explaining to me things I could get myself done. I would also like to thank my parents, relatives and friends for supporting me all along and for having faith in me.”

- Preetham Raj.

"I want to express my sincere gratitude to my parents, family members, and friends for their unwavering love and support. They generously inspired me to pursue my destiny and try out new trajectories in life. I dedicate this accomplishment to them since their faith in me has been the most supporting thing that helped me a lot."

- Prerana Mechineni

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Software Engineering is considered to be an expeditious evolving activity, complex and knowledge comprehensive, where numerous individuals, software teams, and organizations are included in achieving their goals and responsibilities. Software organizations develop their knowledge by solving various challenges when building software products. It results in acquiring and maintaining a creative and consistent advantage in the software development industry [1]. Software companies make continuously development in software products to improvise their quality. Agile software development (ASD) helps in the rapid delivery of the software product with consistent quality to the customer [2].

Software Development gives a challenging aspect on a day-to-day basis while creating it as it depends on software developers' knowledge and creativity. So, there is a need to consider the human activity and knowledge involvement rather than just an engineering aspect. Software development shares a lot of characteristics with knowledge-intensive works, and it also needs lot of knowledge from different domains of both human and computing domains [3].

Knowledge is defined as "justified personal belief that increases an individual's capability to take effective action" [4] [5]. Knowledge was divided into 1) Tacit- which is stored in individuals' memory, or it may result from actions or beliefs. 2) Explicit- knowledge that is stored in the form of data, book, or formula [6] [7]. This has also created the necessity of managing knowledge, so the concept of Knowledge Management (KM) has been adopted. Different industries have studied Knowledge Management (KM) and proposed different concepts and theories [8].

Since most software knowledge is tacit [9] and current practices support knowledge management in agile software development teams. Still, there is a need for frameworks for sharing tacit knowledge in large-scale agile development teams [10] as agile software development depends mostly on tacit knowledge. Moreover, large-scale agile software development is described as the use of agile development principles from large teams to large-scale multi-team projects in a whole organization [11]. Torgeir et al. [11] defined large agile teams as "large-scale agile development teams over 5 teams which consists of more than 50 developers or persons in each team developing together the same product/ project using agile method." As tacit knowledge is hard to extract, we need to invest a lot of time in extracting tacit knowledge by doing software engineering practices like pair programming or co-programming. If there is a

chance, we can introduce agile culture by introducing a coordination mechanism in a structured process to extract and share tacit knowledge rather than micro-managing agile teams.

To introduce a new mechanism or develop an existing mechanism by overcoming its existing challenges, there is a need to study how the current mechanisms are working and what are the challenges faced by the practitioners in the industry for extracting and sharing tacit knowledge in agile software development.

The main aim of this thesis is to explore how tacit knowledge transfer is done between teammates in Agile Software Development. This thesis aims to identify the tacit knowledge transfer mechanism, challenges faced during tacit knowledge transfer, and mitigation challenges to overcome the challenges that are faced during tacit knowledge transfer in agile software development. We have conducted SLR to identify the challenges and mechanisms that are associated with tacit knowledge transfer in agile software development. Later we conducted interviews to gather information regarding the industry's current challenges and the mitigation strategies being followed to overcome these challenges.

1.1 Aim and Objective

Aim: In this present thesis, we focused on how tacit knowledge is being transferred between team members in agile software development and what challenges are faced during tacit knowledge transfer and mitigation strategies to overcome these challenges.

Objective 1: To identify the current tacit knowledge transfer mechanism in agile software development.

Objective 2: To identify challenges in managing tacit knowledge transfer between team members in Agile software development.

Objective 3: To explore mitigation strategies to overcome the challenges faced during tacit knowledge transfer.

1.2 Structure of the thesis

Chapter1: Introduction In this chapter, the author has provided an overview of the thesis along with the aim and objective of the research.

Chapter2: Background and Related work In this chapter, the author has provided the background of the research, and previous work on this study was provided.

Chapter3: Research Design and Methodology In this chapter, the research methods used to conduct this study were explained, and data collection and data

analysis methods are explained.

Chapter4: Results and Analysis In this chapter, an analysis of the data gathered is discussed, and the results obtained are presented along with internal and validity threats mitigated to achieve the results.

Chapter5: Discussions In this chapter, a discussion on how objectives are achieved from the results obtained is presented.

Chapter6: Conclusion and Future work This chapter provides the conclusion of the research based on the research done, along with future work for further research on this topic.

Chapter 2

Background and Related Work

Knowledge has been defined differently by many different authors over the years. The definition of knowledge has been distinguished differently based on knowledge, information, and data [5]. Fahey has stated that knowledge management is nothing interesting if knowledge is not different from data or information. While Churchmans [12] has stated ", knowledge resides in the users and not in the collection." Due to different views on knowledge, Nonaka [13] and Huber [14] have adopted a definition of knowledge that better defines knowledge management in an organizational setting. " Knowledge is a justified belief that increases an entity's capacity for taking effective action" Brown and Duguid [15] have defined that even though the technology and access information are available for the firm, the knowledge cannot be freely circulated among the firm as the knowledge is "sticky".

Nonaka [13] has identified 1) Tacit knowledge and 2) explicit knowledge as the two dimensions of organizational knowledge.

Tacit Knowledge: Tacit knowledge can be referred to as knowledge gained from action, involvement and commitment in a particular field. Polanyi has stated tacit knowledge has both Technical and cognitive elements [16]. Since the tacit knowledge is personalized in the human mind it is difficult to express it in an understanding manner [3]. Tacit Knowledge- It is defined as the phrase "WE KNOW MORE THAN WE CAN TELL". Tacit knowledge is something which is stored in form of skill, imagination or derived from human experience it is difficult to transform this knowledge into documents or blueprints [17] [6].

Explicit Knowledge: it refers to knowledge which is communicated in the form of symbols or natural language [12]. It is something which is studied or acquired in the university that states factual statements [18].
Explicit knowledge can be shared easily.

There is a concept of converting tacit knowledge into explicit knowledge through Implicit knowledge. Implicit knowledge is something that is stored in from imagination or skill but can be converted to documents or data, unlike tacit knowledge. Later implicit knowledge is transformed into explicit knowledge, which helps fulfil the development objective. Knowledge Management (KM) acts as a tool to convert this implicit knowledge to explicit knowledge and helps to transfer this knowledge among the organization [19].

Knowledge Management is the effort the company makes to provide the necessary technology for better sharing or understanding of knowledge, which helps improve the process. There are four types of Knowledge Management Strategies [19].

1. Knowledge Creation: This helps to share and develop knowledge by developing new ideas with the combination of both tacit knowledge and explicit knowledge. Pair Programming is used for this step in agile Software Development [19].

2. Knowledge Storage and retrieval: How organizations keep track of the knowledge in the form of documentation, individual knowledge or database. The organization uses different tools to store this knowledge which is extracted based on past experiences or practices [19].

3. Knowledge Transfer/sharing: Knowledge is transferred within the organization between the employees and teams at different stages of the development of the product. The main challenge in knowledge transfer is not knowing what to share because the individuals don't know what they have knowledge about [19].

4. Knowledge Application: It is defined as the use of knowledge gained for organizational advantage. Knowledge gained from previous sprints is successfully applied in software organizations using the SCRUM framework, review sprint and retrospectives [19].

Agile Software Development: Agile software development relies on communication and collaboration rather than extensive documentation, unlike traditional software development. Agile development —" high-quality adaptive software is developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change" [20].

2.1 Related Work

Koskinen et al. have conducted a study on how agile development can be helpful to extract tacit knowledge without any extra effort and also by overcoming cultural and psychological barriers. And stated that it is difficult to gather tacit knowledge without a proper systematic route, and cultural shift is also important for tacit knowledge management [21].

Agile software development requires a lot of context-dependent knowledge, and it is difficult to reuse the same knowledge for a different product within the same organization. So, it is necessary to analyze the knowledge for successful knowledge management. Tacit knowledge can be translated into explicit knowledge using Knowledge codification to store the knowledge for knowledge management. To share

the knowledge between distributed agile teams' organization, use techniques like pair programming and workshops [22].

The authors Indumini et al. have conducted a literature review on knowledge management in agile development, and they stated that organizations are not aware of which knowledge type should be used in agile software development, and there is a concern about using explicit knowledge in Agile software development even though the organization recognized tacit knowledge as useful knowledge for agile software development [23].

Agile knowledge engineering and management is important in ASD because most of the knowledge required for development is tacit, and it resided in the brain. More than half of the respondents believe learning and sharing are most practised for knowledge sharing, whereas the remaining respondents practice documentation as a way of sharing knowledge. This provides good insight that tacit knowledge plays the main role in agile software development, and there is a need for better knowledge management techniques [24].

2.2 Problem Formulation

Tacit Knowledge is intangible information concerning a skill or area of expertise in any specific field gained through experience [25]. Areas of research and development like Software development require a lot of human creativity and approach being taken to solve a certain problem along with the knowledge of the subject. We have ways of transferring explicit knowledge by documentation. But in the context of tacit knowledge, there are no standard ways of effectively documenting this information so that it can be preserved and transferred to others [24].

While studies like [20] [24] [21] have mentioned most knowledge in agile software development is tacit and mentioned tacit knowledge management is important, there is less discussion about knowledge transfer which is first in the process of knowledge management. As tacit knowledge is not codified, other forms of knowledge transfer than documenting should be explored.

Chapter 3

Research Design and Methodology

3.1 Research questions

RQ1: What are the current tacit knowledge transfer mechanisms used in agile software development?

Motivation: Tacit knowledge transfer is one of the challenging things to do in agile software development. To study the challenges faced in the industry, it is necessary to know what are mechanisms being used currently in the industry.

RQ2: What challenges are faced during tacit knowledge transfer between team members in Agile software development?

Motivation: There are very few researches discussing particular challenges of tacit knowledge transfer, so this research question focuses on gathering challenges faced during tacit knowledge transfer. These challenges are identified using SLR and interviews as research methods.

RQ3: What are mitigation strategies followed to overcome the challenges?

Motivation: Many articles have mentioned the challenges, but there is a discussion on mitigation strategies to overcome these challenges. This RQ focus on gathering these mitigation strategies using the interview as a research method.

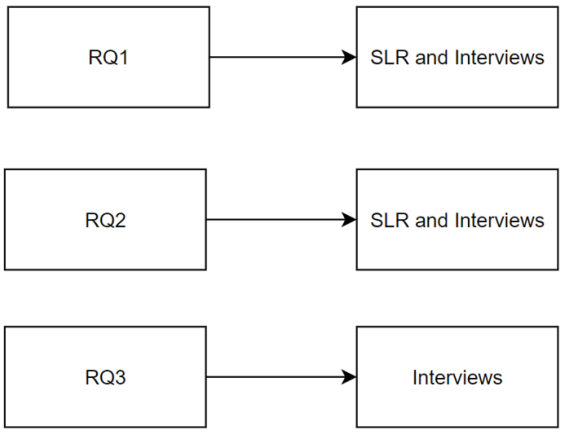

Figure 3.1: Mapping RQ's with Research methods

3.2 Systematic Literature Review

SLR aims to identify existing literature related to the research are [26] SLR can be conducted using Database search and snowballing. SLR is conducted based on primary studies. SLR was considered for this research because it helps to gain knowledge on what challenges and transfer mechanisms are present in the existing literature. Gaining knowledge of this literature results helps the authors to conduct further research without deviating from the research topic.

The main aim of our SLR is to find challenges faced while transferring tacit knowledge between the team members and the transfer mechanisms used in agile software development. The snowballing procedure was considered to conduct SLR, which gives qualitative results for the SLR. SLR provides previous knowledge on what are the challenges faced while transferring tacit knowledge and the mechanism used to transfer tacit knowledge.

3.3 Snowballing Procedure

Snowballing is the autonomous and independent process that is considered as one primary step to perform SLR [27]. Snowballing is defined as a "systematic approach as the snowballing as a procedure formulated here is definitively an alternative to using as a starting point for a systematic literature study instead of always starting by searching different databases" [26]. Relevant papers, which are called start set, was selected to start the snowballing procedure. Later forward and backward snowballing was conducted on the start set to select papers relevant to the study using inclusion and exclusion criteria. Characteristics used in our research for conducting snowballing:

- The scope of the research defines the size of the start set.
- Keywords from the research questions are selected to form a search string, which helps to select start set papers.

3.3.1 Start set identification:

The start set is identified using the below-mentioned characteristics:

- Keywords that are related to the research are identified from the research question.
- Later, these keywords are used to form a search string which helps to search related papers from different databases.
- The above steps are repeated several times to get more suitable keywords for the search string and the final keywords are selected.

Keywords:

- Step1: Tacit knowledge, Knowledge transfer, Agile software development.
- Step2: Challenges, Transfer Mechanism, C Challenges in knowledge transfer.

The above keywords are used to generate different set search strings to identify relevant papers for this research. Inclusion and Exclusion criteria were implemented on the identified papers. Inclusion and Exclusion criteria were discussed below.

- Search String 1: Tacit knowledge, Challenges in tacit knowledge transfer, Agile software development.
- Search String 2: Tacit knowledge, Transfer mechanism, Agile software development, Knowledge Transfer.

Database: Databases like Google scholar, Scopus, and IEEE Explore were the databases considered in this research to identify related papers using a search string. A total of 15 papers have been identified from a database search as part of the start set.

Preliminary start set of articles: Papers from the search string was selected by avoiding unrelated articles based on title, abstract, and keyword. A total of 15 papers are finalized as our starter set.

3.3.2 Inclusion criteria:

- Articles whose titles consist of tacit knowledge and agile software knowledge.
- Articles whose abstract is related to our research.
- Articles whose focus is on challenges in tacit knowledge transfer.
- Articles that are peer-reviewed.

3.3.3 Exclusion criteria:

- Articles whose focus is not on both agile and knowledge transfer.
- Articles which are repetitive.
- Articles which are not peer-reviewed.
- Articles that are not in English.

3.3.4 Forward and backward snowballing iterations:

A total of 3 backward and forward snowballing iterations are conducted based on guidelines mentioned by Wohlin [27]. Backward snowballing is conducted by considering all the papers which refer to the selected article. While forward snowballing is conducted by considering the papers which are cited in the selected articles. Inclusion and exclusion criteria were implemented to identify related papers to our study.

Snowball Sorting:

To check whether all the papers are selected within inclusion/exclusion criteria, Snowball Sorting is implemented. This snowball sorting is performed by cross-checking all the papers identified by the first author by the second author and vice versa. This helps ensure that no paper related to the study is excluded.

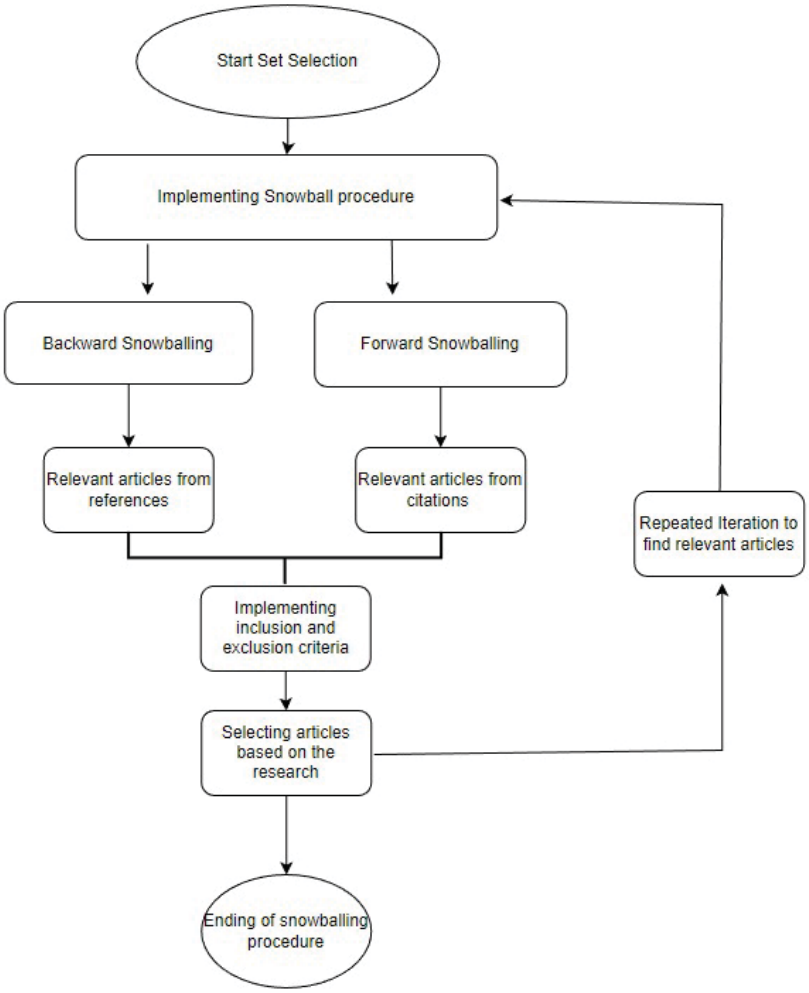

Figure 3.2: Snowballing procedure

3.3.5 Thematic Analysis for SLR

Data is extracted using thematic analysis. Data patterns are identified, analyzed, and reported by analyzing the data using this method [28]. Our study focuses on the challenges faced during tacit knowledge transfer and the transfer mechanisms which are being followed. Other known analysis methods are Narrative analysis and Qualitative comparative analysis. These two methodologies are not suitable for SLR because narrative analysis focuses more on theory building which is based mostly on practitioners' judgment [29], while qualitative comparative analysis focuses on determining casualties in the data [30]. Steps followed in thematic analysis:

1. **1. Familiarizing with the data:** Understanding the raw data extracted from papers is one of the important tasks. So repetitive reading of the data is done to get familiarized with the data. As the first step, challenges in knowledge transfer and mechanisms used to transfer tacit knowledge are identified from the relevant papers [31].

2. **Forming Initial codes:** In this phase, initial codes are generated based on the understanding of the data. Steps followed in generating codes for challenges [31]:

   - Identify the challenges faced.
   - Identify what that challenge means.
   - The above steps are repeated till all the challenges are coded.

   Steps followed in generating codes for the Tacit knowledge Transfer Mechanism:

   - Identify the transfer mechanism.
   - Identity what does mechanism mean
   - The above steps are repeated till all the mechanisms are coded

3. **Translating the codes into themes:** In this phase, all the codes identified are studied repetitively to remove duplicate codes with the same meaning, and later these individual codes are reported as themes [28].

4. **Producing Final Themes:** In this phase, all the codes are mapped to the themes identified. In our research, all the codes identified are reported as individual themes [28].

3.3.6 Quality Assessment

Quality assessment is performed on the start set, and a total of 16 articles have been extracted after performing 3 snowballing iterations. Verification and validation of literature have been done by performing the quality assessment. The quality review was performed on the articles using assessment criteria from the Center for Reviews and Dissemination, 2009 [32]. Both the authors performed assessments individually and later discussed to reduce deviation from the study. A checklist was prepared, and a score was given to each article to check whether the article was valid to include in the study or not. Criteria for conducting the quality assessment are mentioned below:

ID	Criteria	Score
C1	Are research questions formulated based on the research and methods selected?	Y: Formulation of research questions is clear P: The research question formulated is not clear,m N: The research question formulated are not according to the research study

C2	Is the search strategy selected appropriate?	Y: The authors have not deviated from the research topic by selecting multiple databases. P: All the databases are used for selecting articles, but no additional strategies are used N: The authors have deviated from the research topic by implementing a search strategy in multiple databases
C3	Are inclusion and exclusion criteria used clear?	Y: Inclusion and exclusion criteria were clear P: Inclusion and exclusion criteria are not clear N: Inclusion and exclusion criteria were missing
C4	Are research articles selected providing clear information on the research conducted?	Y: Clear information on research data and results was presented in the articles. P: The article has clear research data, but results are not present N: Data is not presented clearly in the article
C5	Are mitigation strategies followed to overcome threats in the research?	Y: Strategies were followed while conducting research on the threats P: Only a few steps were followed to overcome the threats N: No measures were taken to overcome the threats in the research.
C6	Does the conclusion was clearly interpreted by the authors?	Y: The conclusion was clearly stated based on the results obtained. P: Partial conclusion was interpreted. N: No conclusion was provided in the article.

Table 3.1: Represent Quality Assessment

To verify whether the extracted data is qualified, quality scores are extracted by implementing quality assessment criteria. Score of quality assessment are Y(Yes) =1, P(Partial) = 0.5 and N(No)=1. If the quality score is 3 or >3 then the extracted data is considered. Quality assessment scores of SLR are represented in Appendix C.

3.4 Interviews:

Interviews are conducted to collect qualitative data on the research topic from the perspective of participants who have experience working in agile software development and participated in tacit knowledge transfer. Interviews are structured in a

way that interviewers who are students can learn everything about the research topic from interviewees who are experienced professionals. Interviews are one of the efficient ways to collect qualitative data from interviewees on their personal experiences and opinions.

3.4.1 Interview Structure:

Semi-structured interviews provide a way to gain knowledge on what are the challenges and transfer mechanisms, so semi-structured interviews are selected as part of our research. As interviews are conversational, it also provides a way to get answers to the question that may arise based on the response from the interviewees [33].

3.4.2 Selection of Interviewees:

Interviews aim to understand the participants' similar perspectives regarding the research topic. The interviewees selected for conducting interviews are experienced people who have a better understanding of Tacit knowledge and knowl- edge transfer mechanism and who worked in agile development. Semi-structured interviews are conducted as part of the study. An invitation letter was sent to the interviewee with a short description of what our thesis focuses on, what interview questions are regarding, and the interview duration. Based on their availability, interviews are scheduled and conducted . A total of 30+ people have been selected based on their experience and job title, and interview invitations were sent to them through LinkedIn, along with an overview of our thesis. Both authors conducted the interviews together and each interview duration lasted from 45 minutes to 60 minutes. In total, 15 respondents were interviewed.

S.No	Job Title	Experience
I1	Agile coach	10+ years in agile software development.
I2	Software Developer	1-5 years in agile software development.
I3	Scrum Master	10+ years in agile and scrum.
I4	Agile coach	10+ years in agile software development.
I5	Software Developer	5-10 years in agile and scrum.
I6	Scrum Master	1-5 years in agile software development.
I7	Software Developer	0-1 years in agile software development.
I8	Scrum Master	5-10 years in agile software development.
I9	Verification Engineer	1-5 years in agile software development.
I10	Product Owner	5-10 years in agile software development.
I11	Software Developer	0-1 years in agile software development.
I12	Scrum Master	5-10 years in agile software development.
I13	Product Owner	1-5 years in agile software development.
I14	Scrum Master	1-5 years in agile software development.
I15	Product Owner	1-5 years in agile software development.

Table 3.2: List of Interviewee

3.4.3 Interview Questionnaire formulation:

Interview questionnaires are formed based on the results from SLR. The questionnaire is prepared by both authors manually. A pilot interview is conducted to know about the questions which are to be added to the questionnaire and to gain experience in conducting interviews since both the authors are students and have no prior experience in conducting interviews. Since the interview structure is semi-structured, there would be scope for raising more questions that are not formulated. All the interviews are recorded with the permission of the interviewees. So, through the field notes and audio recording, we revised the interview conversation and added the questions if required. Interview questionnaires are presented in Appendix B.

3.4.4 Transcription:

Both authors conducted the interviews together. All the interviews which are conducted are recorded interviewee's permission. All the audio files recorded are transcribed to text using Descript software. For transcribing, all the recordings are given a unique name, so none of the recordings is missed without transcribing. The transcribed text was stored in word format to analyze the data.

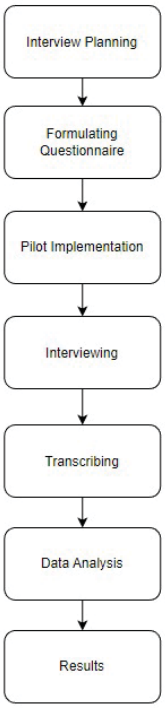

Figure 3.3: Interview Design

3.4.5 Basis For Rejecting Other Research Methods

Systematic literature reviews and interviews were considered research methodologies for this research study as they both are exploratory in nature and no relationships

are measured. Other research methods like surveys, experiments, and case studies were rejected.

The survey derives its results or conclusions depending on the abundant data results and factors. In contrast, our research study needs data from practitioners with real-time experience and face-to-face conversations for in-depth results. A case study derives similar results to the survey, which might not give a real insight into deriving the challenges of this research study. An experiment study mainly measures independent and dependent variables affecting the environment's results [34]. This research is based on extracting the derived challenges and real-time, i.e., new period challenges. Hence, an experiment has not been considered for this study [34].

3.5 Thematic Analysis of Interviews:

Braun & Clarke [28] approach is considered to conduct a thematic analysis of the interviews.

- **Familiarizing with the results:** The first step is to get familiarized with the data and get a better understanding of the data. All the recorded interviews are transcribed, and repeated study of the data is conducted. Braun [28] mentioned that the first step is the most important as it may lead to deviated results if we miss any important information. To avoid this mistake, audio transcribed by the first author was cross verified by the second author to make sure no data was missed and vice versa. Once all the data is transcribed, the authors colour code the challenges and mechanisms identified from the transcribed data.

- **Generating initial codes:** Initial codes can be generated manually or with the help of software application tools like NVivo, MAXQDA, and Atlas. We have opted to analyze and code the data manually to avoid missing any important information, as we are collecting information from the interviewee's opinions based on their experience. The authors conducted an analysis and colour-coded the data individually. Once all the colour-coded data is converted into initial codes individually, the analyzed data is exchanged for further verification.

- **Searching for Themes:** In this third phase, all the codes are studied further to find standard links between the codes, which can be categorized into themes. Boyatzis et al. [35] stated that all the codes should be structured and focused on the same goal to generate good themes. Since all the codes identified from the data are distinct and unique, we presented individual codes as themes.

- **Producing Final Themes:** Final themes are produced, and all the findings are related to the goal of the thesis and presented in the results and analysis section.

3.5.1 Narrative Analysis

Narrative analysis is considered to summarize open-ended questions where interviewees express their opinion [29]. Human perceptions or opinions on the research topic

and summarized into story points. Both qualitative and quantitative data gathered in both SLR and interviews can be analyzed using narrative analysis. Data can be organized in a structured way by analyzing and visualizing the data by interpreting human perceptions, and it also increases the understandability of the data [29].

In this research, interview question Q6 which focuses on collecting effective mechanisms among all the mechanisms mentioned to transfer the tacit knowledge is analyzed using narrative analysis.

Chapter 4

Results and Analysis

4.1 Snowball Sampling Results

A total of 15 papers have been selected from google scholar as a starting set.

Article ID	Starts set Article
P1	Ryan, Sharon, and Rory V. O'Connor. "Acquiring and sharing tacit knowledge in software development teams: An empirical study." Information and Software Technology 55.9 (2013): 1614-1624. [36]
P2	Kavitha, R. K., and MS Irfan Ahmed. "A knowledge management framework for agile software development teams." 2011 International Conference on Process Automation, Control and Computing. IEEE, 2011. [37]
P3	Ryan, Sharon, and Rory V. O'Connor. "Social interaction, team tacit knowledge and transactive memory: Empirical support for the agile approach." (2012). [38]
P4	Juárez-Ramírez, Reyes, Rafael Pimienta-Romo, and Violeta Ocegueda-Miramontes. "Using social networks for integrating a tacit knowledge repository to support the performance of software development teams." International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making. Springer, Berlin, Heidelberg, 2013. [39]
P5	Heredia, Alberto, et al. "Interactive knowledge asset management: Acquiring and disseminating tacit knowledge." Journal of Information Science and Engineering 29.1 (2013): 133-147. [40]
P6	Chau, Thomas, and Frank Maurer. "Knowledge sharing in agile software teams." Logic versus approximation. Springer, Berlin, Heidelberg, 2004. 173-183. [41]
P7	Bjørnson, Finn Olav, and Kathrine Vestues. "Knowledge sharing and process improvement in large-scale agile development." Proceedings of the Scientific Workshop Proceedings of XP2016. 2016. [10]

P8	Camacho, José Jairo, J. M. Sanches-Torres, and Ernesto Galvis-Lista. "Understanding the process of knowledge transfer in software engineering: a systematic literature review." The International Journal of Soft Computing and Software Engineering. Special Issue: The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [42].
P9	Ghobadi, Shahla. "What drives knowledge sharing in software development teams: A literature review and classification framework." Information & Management 52.1 (2015): 82-97. [43]
P10	Zahedi, Mansooreh, Mojtaba Shahin, and Muhammad Ali Babar. "A systematic review of knowledge sharing challenges and practices in global software development." International Journal of Information Management 36.6 (2016): 995-1019. [44]
P11	Foos, Ted, Gary Schum, and Sandra Rothenberg. "Tacit knowledge transfer and the knowledge disconnect." Journal of knowledge management (2006). [45]
P12	Chau, Thomas, Frank Maurer, and Grigori Melnik. "Knowledge sharing: Agile methods vs. tayloristic methods." WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.. IEEE, 2003. [46]
P13	Ouriques, Raquel Andrade Barros, et al. "Knowledge management strategies and processes in agile software development: a systematic literature review." International journal of software engineering and knowledge engineering 29.03 (2019): 345-380. [19]
P14	Aurum, Aybüke, Farhad Daneshgar, and James Ward. "Investigating Knowledge Management practices in software development organisations–An Australian experience." Information and Software Technology 50.6 (2008): 511-533. [9]
P15	Koskinen, Kaj U., Pekka Pihlanto, and Hannu Vanharanta. "Tacit knowledge acquisition and sharing in a project work context." International journal of project management 21.4 (2003): 281-290. [47]

Table 4.1: Start Set of Papers

4.1.1 First Iteration

As part of the first iteration of snowballing, a total of 258 articles have been reviewed. Out of 258 articles, articles focusing on tacit knowledge transfer, agile software development, and challenges are analyzed, and a total of 6 articles have been selected. The remaining 252 articles are rejected based on title and abstract. Later, inclusion and exclusion criteria are applied to avoid any repetitive or non-English articles. The results of the first forward snowballing iteration are:

Article ID	**First Literature of forward snowballing**
P16	Takpuie, Deon, and Maureen Tanner. "Investigating the characteristics needed by scrum team members to successfully transfer tacit knowledge during agile software projects." Electronic journal of information systems evaluation 19.1 (2016): pp36-54 [48].
P17	Heeager, Lise, and Peter Axel Nielsen. "Agile software development and the barriers to transfer of knowledge: an interpretive case study." Scandinavian conference on information systems. Springer, Berlin, Heidelberg, 2013 [49].
P18	Dreyer, Hanna, and Martin G. Wynn. "Tacit and Explicit Knowledge in Software Development Projects: A Combined Model for Analysis." International Journal on Advances in Software 9.3/4 (2016): 154-166 [50].
P19	Spraggon, Martin, and Virginia Bodolica. "Collective tacit knowledge generation through play: Integrating socially distributed cognition and transactive memory systems [51]." Management Decision (2017).
P20	Mtsweni, Emmanuel Samuel, and Nehemiah Maveterra. "Issues affecting application of tacit knowledge within software development project." Procedia computer science 138 (2018): 843-850 [52].
P21	Razzak, Mohammad Abdur, and Rajib Ahmed. "Knowledge sharing in distributed agile projects: Techniques, strategies and challenges." 2014 Federated Conference on Computer Science and Information Systems. IEEE, 2014 [1].

Table 4.2: First Iteration Snowballing Articles

4.1.2 Second Iteration:

As part of the second iteration, a total of 187 articles have been studied. Out of 87, 7 articles that are related to our study have been identified using inclusion and exclusion criteria. The remaining 180 articles are rejected based on title and abstract. The results of the second iteration are presented below:

Article ID	**Second Iteration of snowballing**
P22	Castaneda, Delio Ignacio, and Paul Toulson. "Is it possible to share tacit knowledge using information and communication technology tools?." Global Knowledge, Memory and Communication (2021 [53]).
P23	Idrus, Hariaty Mohd. "Tacit Knowledge in Software Testing: A Systematic Review." 2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS). IEEE, 2019 [54].
P24	Dreyer, Hanna. Tacit Knowledge in a Software Development Project. Diss. University of Gloucestershire, 2018 [55].

P25	Mtsweni, Emmanuel Samuel, and Nehemiah Maveterra. "Issues affecting the application of tacit knowledge within a software development project." Procedia computer science 138 (2018): 843-850 [52].
P26	Gervigny, Marie Liliane Isabelle, and Soulakshmee D. Nagowah. "Knowledge sharing for agile distributed teams: A case study of Mauritius." 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS). IEEE, 2017 [56].
P27	Ryan, Sharon, and Rory V. O'Connor. "Acquiring and sharing tacit knowledge in software development teams: An empirical study." Information and Software Technology 55.9 (2013): 1614-1624 [36].
P28	Razzak, Mohammad Abdur, and Rajib Ahmed. "Knowledge sharing in distributed agile projects: Techniques, strategies and challenges." 2014 Federated Conference on Computer Science and Information Systems. IEEE, 2014 [1].

Table 4.3: Second Iteration Snowballing Articles

4.1.3 Third Iteration:

In this iteration, a total of 142 articles have been studied. Out of 142, a total of 3 articles have been selected by implementing inclusion-exclusion criteria. The remaining 139 articles are rejected based on full text, abstract, and title.

Article No	**Third Literature of forward snowballing**
P29	Mtsweni, Emmanuel Samuel, and Nehemiah Mavetera. "Soft Issues that Limit Sharing of Tacit Knowledge within Software Development Project Teams." 2019 IEEE AFRICON. IEEE, 2019 [57].
P30	Buunk, Iris, Colin F. Smith, and Hazel Hall. "Tacit knowledge sharing in online environments: Locating 'Ba'within a platform for public sector professionals." Journal of Librarianship and Information Science 51.4 (2019): 1134-1145 [58].
P31	De Brito, Maylon F., et al. "Knowledge transfer in a management process for outsourced agile software development." Proceedings of the 50th Hawaii International Conference on System Sciences. 2017 [59].

Table 4.4: Third Iteration Snowballing Articles

4.1.4 Thematic Analysis results of SLR

Code	**Challenges**	**Explanation**
C1	Lack of verbal knowledge	Less usage of verbal communication between teammates makes it difficult. [P1][P5][P27]

C2	Lack of relationship	Lack of interaction between the teammates. [P2][P29][P9]
C3	Personality	lack of personalities which is compatible with other teammates [P4][P30][P18]
C4	Lack of critical thinking	Unable to articulate or visualize tacit knowledge to make others understand the concept [P3][P30]
C5	Lack of human orientation	Not being a teammate to whom others can easily approach [P6][P31][P13][P14]
C6	Socialization or communication	Unable socialize with other team members[P8][P31][P27][P17]
C7	Change management	Unable to adopt the knowledge management tools that were changed and adapted over time. [P10][P28][P9][P8]
C8	Time Management	No proper management to identify tacit knowledge[P10][P26][P7]
C9	Time Zones	Distributed teams have difficulty having common working hours to share knowledge [P24][P22][P6][P17]
C10	Language barriers	Unable to understand the language[P6][P5][P14]
C11	Informal communication Language	No usage of technical terms to share the knowledge[P7][P9][P12][P4]
C12	Misunderstanding	Unable to understand or visualize knowledge shared by another team member [P11][P21][P10][P2]
C13	Visualization	Unable to visualize for making other to understand[P13][P27]
C14	Lack of Technology	Unavailability of knowledge management systems and technologies such as Jira that can support and ease the process of sharing tacit knowledge[P15][P25][P13]
C15	Lack of Information	having less information on the knowledge to make others understand or document the knowledge [P17][P27]
C16	Time and resources	If you lack time to identify other individuals or groups with specific knowledge it is difficult to share and also having no resources to share knowledge is a barrier[P18][P19]

C17	Organizational Culture	Corporate culture and the organizational design support the knowledge transfer practices[P11][P12][P20]
C18	Individual skills	Individual skill of not socializing, no good coordination with other teammates [P11][P13][P7][P16][P17]
C19	Motivation and willingness	Not willing to share or receive knowledge[P18][P4][P1][P20][P22]
C20	Management Style	Distributed teams have difficulty having common working hours to share knowledge [P16] [P18][P25]
C21	Trust	Not trusting the person who shares the knowledge or to whom you are sharing [P28][P27][P23][P14]

Table 4.5: Challenge faced during Tacit knowledge transfer

Code	**Explanation**	**Mechanism**
M1	Repositories	online database that systematically captures, organizes and categorizes knowledge-based information[P4][P7][P19]
M2	Pair programming	Pair programming is an Agile software development technique originating from Extreme programming (XP) in which two developers team together on one computer.[P10]
M3	Version control	the process by which different drafts and versions of a document or record are managed[P5][P14][P22]
M4	Screen Sharing	creating boards like Trello to share knowledge[P24][P3O][P3]
M5	Daily scrum	Daily meeting to share the work done on daily basis[P7][P11][P18][P27]
M6	Weekly sprints status	Weekly meeting to share the work done on weekly basis[P25][P9][P17][P29]
M7	Common chat room	Chat rooms for whole team to share what they have learnt[P21][P14][P6][P8]
M8	Technical forum	forums within the organization to share technical aspects[P16][P20][P31]
M9	Discussion forum	Forum to discuss common interested topics [P22][P31][P4][P17]
M10	Electronic board	Sharing board for transfer of knowledge digitally [P11][P4][P23]

M11	Online Conference	Online conference for distributed teams[P6][P18]
M12	Wiki	a smart-text editor that doubles as a communication machine where you draft, edit, and chat all in one place [P10][P3][P27]

Table 4.6: Mechanisam used to Transfer Tacit knowledge

4.1.5 Thematic Analysis results of Interviews

Targeted audiences are contacted through LinkedIn to schedule interviews. Once the audience accepts the interview request, interviews are conducted according to their availability. A total of 15 interviews were conducted. The interview questionnaire consists of both open-ended and closes ended questions. Open-ended focus on demographics, and close-ended focused on collecting information to answer the research questions. Thematic analysis was performed to obtain the results.

Code	**Challenge**	**Explanation**
C1	Technical Terminology	Lack of knowledge on technical terms [I1][I4][I11]
C2	Attitude	Other person is ready to listen to you when you are sharing ideas. [I12][I14]
C3	Less Experience	Having less working experience makes people to difficult to accept even if you have good knowledge on the topic. [I2][I8]
C4	Language Barrier	Working around the globe as a part of virtual team makes it difficult as English slang differs from region to region. [I9][I5][I1]
C5	Time management	Working in distributed makes it difficult to find time for having knowledge sharing sessions. [I6][I15][I10]
C6	Lack of Knowledge	Lack of knowledge on what is going on or what is the problem. [I6][I9][I7]
C7	Verbal Knowledge	Lack of knowledge of verbal words which makes it easy for the others to understand easily. [I8][I10][I14]
C8	Different Perspective	All the team members may not have knowledge of all other domains, so it makes it difficult to share knowledge from their perspective which makes it easy to understand for them.[I1][I5]

C9	Open to feedback	Some of the members may not be willing to accept feedback on the knowledge they shared. [I3][I7][I10]
C10	Willingness	How willing you are to share or receive knowledge. [I2][I8][I9][I13]
C11	Time Zone	Working at different times results in a lack of communication. [I4][I11]
C12	Lack of Competence	Having knowledge of the process but doesn't have knowledge of the technical words which makes it easy for other members to understand. [I1][I7][I8]

Table 4.7: Challenges faced during Tacit knowledge transfer

Code	**Transfer Mechanisam**	**Explanation**
M1	Screen Sharing	Presenting the knowledge through the presentation by screen sharing. [I6][I11][I15]
M2	Audio/ Video Clips	Creating audio/ video clips so that other team members can receive/ share knowledge. [I2]
M3	Scrum Meeting	Dedicated scrum calls to share knowledge among team members before starting of the sprint. [I8][I12][I1]
M4	Daily Scrum Calls	Daily calls to discuss what they have done personally the previous day. [I5][I8][I14]
M5	Forums	Dedicated forums to discuss knowledge gained on personal experience. [I4][I7][I9]
M6	One-on-One meetings	Having meetings with team members who work on the same domain. [I3][I6][I10][I12]
M7	Knowledge transfer sessions	Dedicated knowledge transfer sessions through presentations which are later documented for further usage.[I11][I13][I8]
M8	Story Telling	Taking up an example or real-life experience for better understanding. [I4][I10][I12]

Table 4.8: Transfer Mechanism

Code	**Mitigation Strategies**	**Explanation**
Mi1	Basic knowledge	Since the agenda for the meeting is presented before, it is better to have a basic understanding of the things which are going to be discussed [I7][I10]
Mi 2	Organized Meetings	Better to have meetings organized beforehand for a better flow of knowledge between team members and also to gain knowledge on what the team members are up to on their respective tasks.[I4][I5][I9][I13][I14]
Mi 3	Face to Face Meetings	Face to face meeting either personally or virtually motivate people to try new things, and it is the better way to guide others. [I3][I6][I11][I13][I15]
Mi 4	T- Shaped	Where certain people are assisted to particular tasks to share knowledge which avoids irrelevant roles to take part in knowledge sharing. [I2][I7][I9]
Mi 5	Full Agile teams	It is better to have agile teams as they scrum master, product owner who takes responsibility to organize knowledge transfer session based on the necessity of the project. [I1][I6][I10]
Mi 6	Online whiteboarding	Space to share knowledge gained from personal experiences [I3][I8][I12][I13]
Mi 7	Retrospective Tools	Use to retrospective tools like Jira to share knowledge on problems that the team is trying to solve. [I5][I4][I7]

Table 4.9: Mitigation Stratagies

4.1.6 Results of Narrative Analysis of Interview

Narrative Respondents Information

First, five close-ended questions focus on gathering information about the interviewee's job role, experience, and whether they worked on agile methodology.

Question Q1 focuses on the role of the interviewee in the organization. A few roles like scrums masters, agile coaches, software developers, and product owners are considered while selecting the interviewees. Out of 15 interviewees, 33.3% are scrum Masters, 13.3% are Agile coaches, 26.6% are software developers, 6.6% are Verification Engineers, and 20% are Product Owners. Multiple roles of the interviewee are also considered and role in which they have worked is presented in the results.

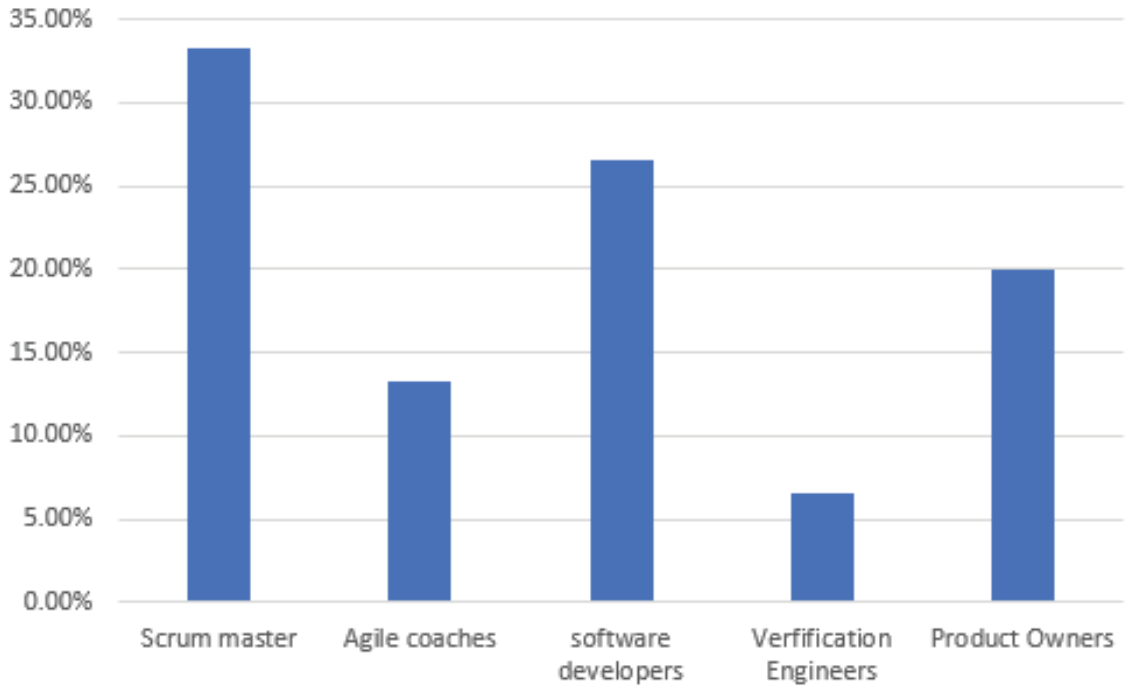

Figure 4.1: Job Titles

Question Q2 focuses on the number of years of experience in agile software development. 0-1, 1-5, 5-10, and 10+ years are the four categories into which years are divided. Out of 15 respondents, 13.33% have experience between 0-1, 40% respondents have experience between 1-5 years, 26.66% has experience between 5-10 years, and 20% has experience more than 10+ years.

No of years of Experience	No of Interviewees
0-1 years	2
1-5 years	6
5-10 years	4
10+ years	3

Table 4.10: No of Years of experience

Question Q3 focuses on whether the interviewees have experience working in Agile software development. All 15 have experience in agile either in the form of working in the agile development life cycle or through agile coaching.

S. No.	Methodologies	No. of interviews
1	Agile methodologies	13
2	Agile and Scrum	2

Table 4.11: Methodologies

Question Q4 focuses on whether the interviewees are part of tacit knowledge sharing in agile software development. Fortunately, all the interviewees are part of a tacit knowledge sharing in both knowledge receiving and knowledge transferring.

Question Q5 focuses on the type of teams they have worked with. 8 interviewees worked mostly with distributed teams, and 4 interviewees worked only with virtual teams, and 3 interviewees worked with hybrid teams.

S. No.	Type of teams	No. of interviewees
1	Distributed teams	8
2	Virtual teams	4
3	Hybrid teams	3

Table 4.12: Type of Teams

Question Q6 focuses on what is the effective mechanism to transfer tacit knowledge in agile software development based on their experience.

Results Interviewees were asked to give their opinion on the effective mechanism of all the mechanisms they mentioned. Narrative analysis is performed and results are presented below. Effective mechanisms based on interviewee experience are:

- Every team should have two meetings every week during the sprint. The first meeting deals with assigning tasks to each person and knowledge transfer regarding that task to that particular. In the second, all the team members gather together and share the tacit knowledge that they have gained while working on the task; this allows all the team members to gain knowledge on the whole project.

- Technical forums or Discussion forums: these forums allow the developers to share their knowledge with others who are working in the same domain and to gain knowledge. Using these forums converts tacit knowledge to explicit knowledge, which can be saved for further use.

- Training sessions: Tacit knowledge is something that is not documented, but it is present in the brain, which helps to solve the problems being faced. Sometimes during tacit knowledge transfer sessions, we may miss some of the knowledge which needs to be shared, which may affect the sprint dead, so frequent training sessions should be conducted to make sure necessary knowledge is transferred by responsible persons like scrum masters, product owners, etc.

  Some of the Interviewee's opinions are:

  - "Based on the situation, it is necessary to prioritize the tasks and share knowledge on those tasks rather than sharing the knowledge randomly on all the tasks."

  - "For distributed teams and virtual teams, I personally think scrum meetings are not so effective because I need to learn about their work routine or task briefly to understand their knowledge terminology due to language barriers."

  - "Companies should invest some budget and time into persons who are responsible for tacit knowledge transfer by providing an architecture for tacit knowledge transfer."

4.1.7 Threats to Validity of SLR

Internal Validity Internal validity is the attributes that influence the procedure followed and results achieved for the research [60]. Search string formulation is one of the procedures followed, which can be influenced by internal validity. A description of the internal validity of string formulation is presented below.

- Formulating the search string: String formulation is one of the crucial attributes that can affect the focus of the study. The related articles that are collected will be inappropriate if the string formulated is unclear. To avoid these threats, the string was formulated fairly by doing a rigorous study. And both authors and supervisor have verified the string formed.

External Validity Conducting SLR increases the risk continuously, as there is a chance of not finding enough data which is related to the study. This can also

affect the result of the thesis as relevant data is missing. For conducting this study, we have focused on articles which has information about tacit knowledge transfer mechanism and tacit knowledge transfer challenges in agile software development. To make sure all the articles related to the study are identified, we have conducted three snowball iterations while conducting SLR.

Construct Validity: For the snowball study procedure, construct validity mentions the valid information or data about the existence of the startle factors, verifying if the study is valid to apprehend its aims and objectives. The construct validity threats were minimalized by having a detailed and planned approach for the formulated questions correspondingly. Obtaining a good start set of articles was one of the major threats to a snowballing approach for SLR. In order to obtain a good start set of articles, Wohlin [27] were considered, followed by a change in the selection of the data. It reduces the risk of pertaining irrelevant studies for the research. Adding to it, during the selection of start of papers, to mitigate the risk of obtaining similar or relatable articles, identical articles of the authors' are excluded. Finally, the supervisor review was taken to make sure the start set selected is on the right path and to validate the snowballing iteration to avoid any uncertainties.

Conclusion Validity: In the conclusion validity, the credibility of the outcome of the research was handled to ensure the results are leading to an appropriate conclusion. During the analysis of primary studies, complex inclusion criteria were constructed and analyzed to make sure all the challenges and transfer mechanism are not omitted.

4.1.8 Threats of validity of Interviews

Internal Validity: Attributes that influence the procedure and results are called internal validity threats. An interview questionnaire is one of the processes which can be affected while conducting interviews.

Interview formulation: Interview questions are crucial for conducting the interviews. Formulating a questionnaire that does not align with our objectives results in not collecting the data required for the study. To make sure all the research questions are aligning with the research goal, the questionnaire prepared is reviewed thoroughly by both supervisor and authors. Pilot interviews are conducted, and questionnaires are modified based on the results from the pilot interview.

External Validity: External validity threats are related to result generalization. To mitigate this threat, sample populations from different organizations varying different sizes are selected. The formulated questionnaire was used while conducting interviews among the selected population to gather generalized opinions on the research topic.

Construct Validity: The threats in the construct validity appear in the context of what is being measured and analyzed [60]. The threats to construct validity

arise because of inadequate knowledge in revising and inspecting the questionnaire before it is shared. The formulation of the questionnaire was continuously revised by the supervisor. From the feedback given by the supervisor, the questions were formulated, and a pilot interview is done before finalizing the interview questions.

Conclusion Validity: Quality and credibility are the two aspects in which conclusion validity threats can occur [60]. To identify the three contexts in the responses we received through the interview, we have considered thematic analysis and narrative analysis for the interview questionnaire. As our interview consists of both open and close-ended questions equally, we have made a narrative analysis for the open-ended questions and thematic analysis for open-ended questions as it consists of challenges, mitigation strategies, and mechanisms used for tacit knowledge transfer in agile software development.

Chapter 5

Discussion

5.1 Discussions

Tacit knowledge plays a crucial role in agile software development. So, this thesis focused on what are tacit knowledge transfer mechanisms which are being used in the industry and what are challenges faced during knowledge transfer, and their mitigation strategies. In this section results of all the research questions are discussed.

RQ1: What are the current tacit knowledge transfer mechanisms used in agile software development?

For this research question, we have conducted both SLR and interviews to identify the tacit knowledge transfer mechanisms which are being used currently. From Slr, a total of 12 transfer mechanisms are identified. From Interviews, a total of 8 transfer mechanisms are identified.

From the SLR, one of the mechanisms is Common chat rooms. These chat rooms provide access to all the members of the company to discuss topics that they are interested in with other members. These discussions provide an opportunity to share the ideas or knowledge they have gained based on their personal experience. Storytelling is one of the mechanisms which is being followed in the industry based on interviewee opinions. Storytelling is a form of sharing knowledge with others by telling it in the form of a story by taking real-life examples. This provides the necessary understanding to the people, even those who don't have much knowledge of technical terms.

The author Indumiti et al., [23] mentioned that explicit knowledge is mostly used in agile software development even though tacit knowledge is considered important. To support this finding, many of the interviewees mentioned all the tacit knowledge sessions should be recorded, or common platforms like chat rooms and wikis should be saved, which can be used for further reference. This is nothing but tacit knowledge is being converted to explicit knowledge.

There are a few interesting findings about the transfer of tacit knowledge mechanisms which consists of continuous scrum meetings, daily scrum calls, and day-to-day forums among the teams, identified both in literature and interview.

RQ2: What are the challenges faced during tacit knowledge transfer between team members in Agile software development?

This research question focuses on challenges faced by practitioners during tacit knowledge transfer in agile software development. Levy et al., [21] discussed that challenges like cultural barriers should be overcome to extract tacit knowledge, and most of the practitioners believe sharing is the efficient way to transfer tacit knowledge in agile software development. To answer this research question, we have conducted both SLR and interviews.

A total of 21 challenges have been identified in SLR, and a total of 12 challenges have been identified from the interview.

The common challenges they have faced are time zone constraints, language constraints, and perceptions of one own team members contradicting the other team members. The absence of knowledge sharing of perceptions or opinions is one of the main challenges faced among the team members. The major findings from the literature and interviews were management style and lack of competence respectively. Many of the interviewees worked in distributed teams. So, the interpretation of the concept or the software product done by a practitioner has been challenging for the other practitioner to understand and execute the same or mutual visualization of that concept or the software product.

Moreover, the literature and interview results indicate that one of the most challenging parts while transferring tacit knowledge is the lack of critical thinking with human orientation. In simple words, the team members were unable to explain the visualization of the concept, which led to a lack of information while delivering any project. The improper explanation or approach of a team member also results in being a challenging part of tacit knowledge transfer.

RQ3: What are mitigation strategies followed to overcome the challenges?

This question focuses on mitigation challenges to overcome challenges faced during tacit knowledge transfer in agile software development.

A total of 7 mitigation strategies have been identified. Most interviewees responded that having pre-planned or organized meetings is necessary for the successful transfer of tacit knowledge in agile software development. But these mitigation strategies also have a few disadvantages.

For example, a T-shaped strategy is followed by one company, in which a person is assigned to a particular who is responsible for specific knowledge transfer to complete the task. This helps to save the time of the developer by getting knowledge from that person instead of waiting for knowledge transfer. This also helps to complete tasks in the assigned time. But the disadvantage of this strategy is that other team members don't gain knowledge of what is going on in the project, which limits

their knowledge to specified tasks or the domain of the tasks.

For distributed teams, the time zone is one of the biggest challenges to overcoming these strategies, like online white-boarding and retrospective tools are used. But using these strategies may result in missing information as some of the knowledge cannot be converted into text.

Contradicting many interviewees' belief of having regular sprint meetings for better tacit knowledge transfer, one of the interviewees stated that sprint meetings are not so effective as they take a lot of time than expected due to language barrier and time zone differences.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

Most of the knowledge in agile software development is tacit, so there is a need to study tacit knowledge management works. Our contribution to this research is to provide an analysis of all the challenges faced while tacit knowledge transfer, to find out all the transfer mechanism used in the industry, and to identify mitigation strategies to overcome the challenges.

From SLR and interviews, many challenges, mechanisms, and mitigation strategies are identified. Based on the interviews, we can determine that distributed teams face most of the challenges compared to other agile teams. We can conclude that even though tacit knowledge plays a main role in agile software development, it is necessary that all the tacit knowledge should be recorded and documented for further usage, which avoids ambiguity in visualizing and interpreting any concept or software product.

6.2 Future Work

In this research, we have identified some of the challenges faced, particularly in some agile teams. Further research can be conducted analyzing tacit knowledge transfer in particular, and a framework can be created for better knowledge transfer in agile development teams. And also, research can be done separately in multinational and startup companies. This helps to understand how tacit knowledge is transferred in those particular companies. Moreover, further research can be done on verifying the outcomes of the research by implementing different tacit knowledge transfer mechanisms in the individual agile teams and measuring the impact on teams' performance with the help of Key Performance Indicators.

References

[1] Mohammad Abdur Razzak and Rajib Ahmed. Knowledge sharing in distributed agile projects: Techniques, strategies and challenges. In *2014 Federated Conference on Computer Science and Information Systems*, pages 1431–1440. IEEE, 2014.

[2] Laurence Prusak and Thomas Davenport. Working knowledge: how organizations manage what they know. 1998.

[3] Yunwen Ye. Supporting software development as knowledge-intensive and collaborative activity. In *Proceedings of the 2006 international workshop on Workshop on interdisciplinary software engineering research*, pages 15–22, 2006.

[4] Maryam Alavi and Dorothy Leidner. Knowledge management systems: issues, challenges, and benefits. *Communications of the Association for Information systems*, 1(1):7, 1999.

[5] Maryam Alavi and Dorothy E Leidner. Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS quarterly*, pages 107–136, 2001.

[6] Ikujiro Nonaka, Ryoko Toyama, and Akiya Nagata. A firm as a knowledge-creating entity: a new perspective on the theory of the firm. *Industrial and corporate change*, 9(1):1–20, 2000.

[7] Jia Hao, Qiangfu Zhao, Yan Yan, and Guoxin Wang. A review of tacit knowledge: Current situation and the direction to go. *International Journal of Software Engineering and Knowledge Engineering*, 27(05):727–748, 2017.

[8] Viviane Santos, Alfredo Goldman, and Cleidson RB De Souza. Fostering effective inter-team knowledge sharing in agile software development. *Empirical Software Engineering*, 20(4):1006–1051, 2015.

[9] Aybüke Aurum, Farhad Daneshgar, and James Ward. Investigating knowledge management practices in software development organisations–an australian experience. *Information and Software Technology*, 50(6):511–533, 2008.

[10] Finn Olav Bjørnson and Kathrine Vestues. Knowledge sharing and process improvement in large-scale agile development. In *Proceedings of the Scientific Workshop Proceedings of XP2016*, pages 1–5, 2016.

[11] Torgeir Dingsøyr and Nils Brede Moe. Towards principles of large-scale agile development: A summary of the workshop at xp2014 and a revised research agenda. In *Agile Methods. Large-Scale Development, Refactoring, Testing, and*

Estimation: XP 2014 International Workshops, Rome, Italy, May 26-30, 2014, Revised Selected Papers 15, pages 1–8. Springer, 2014.

[12] C West Churchman. The design of inquiring systems basic concepts of systems and organization. 1971.

[13] Ikujirō Nonaka, Ryoko Toyama, and Toru Hirata. *Managing flow: A process theory of the knowledge-based firm*. Springer, 2008.

[14] George P Huber. Organizational learning: The contributing processes and the literatures. *Organization science*, 2(1):88–115, 1991.

[15] John Seely Brown and Paul Duguid. Knowledge and organization: A social-practice perspective. *Organization science*, 12(2):198–213, 2001.

[16] M Polanyi. The tacit dimension, doubleday and company inc. *Garden city New York*, 1966.

[17] Kristian Kreiner. Tacit knowledge management: the role of artifacts. *Journal of knowledge management*, 2002.

[18] Kaj U Koskinen. *Management of tacit knowledge in a project work context*. Finnish Academies of Technology, 2001.

[19] Raquel Andrade Barros Ouriques, Krzysztof Wnuk, Tony Gorschek, and Richard Berntsson Svensson. Knowledge management strategies and processes in agile software development: a systematic literature review. *International journal of software engineering and knowledge engineering*, 29(03):345–380, 2019.

[20] Sridhar Nerur, RadhaKanta Mahapatra, and George Mangalaraj. Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5):72–78, 2005.

[21] Meira Levy and Orit Hazzan. Knowledge management in practice: The case of agile software development. In *2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*, pages 60–65. IEEE, 2009.

[22] Siva Dorairaj, James Noble, and Petra Malik. Knowledge management in distributed agile software development. In *2012 Agile Conference*, pages 64–73. IEEE, 2012.

[23] Udeshika Indumini and S Vasanthapriyan. Knowledge management in agile software development-a literature review. In *2018 National Information Technology Conference (NITC)*, pages 1–7. IEEE, 2018.

[24] Amitoj Singh, Kawaljeet Singh, and Neeraj Sharma. Agile knowledge management: a survey of indian perceptions. *innovations in Systems and Software Engineering*, 10(4):297–315, 2014.

[25] Iris Buunk, Hazel Hall, and Colin Smith. Tacit knowledge sharing: the determination of a methodological approach to explore the intangible. 2017.

[26] David Budgen and Pearl Brereton. Performing systematic literature reviews in software engineering. In *Proceedings of the 28th international conference on Software engineering*, pages 1051–1052, 2006.

[27] Claes Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, pages 1–10, 2014.

[28] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.

[29] Brian T Pentland. Narrative methods in collaborative systems research. In *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences. 1999. HICSS-32. Abstracts and CD-ROM of Full Papers*, pages 9–pp. IEEE, 1999.

[30] Andrew Atherton and Peter Elsmore. Structuring qualitative enquiry in management and organization research: A dialogue on the merits of using software for qualitative data analysis. *Qualitative research in organizations and management: An International Journal*, 2(1):62–77, 2007.

[31] Daniela S Cruzes and Tore Dyba. Recommended steps for thematic synthesis in software engineering. In *2011 International Symposium on Empirical Software Engineering and Measurement*, pages 275–284. IEEE, 2011.

[32] Barbara Kitchenham, O Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, 51(1):7–15, 2009.

[33] Daniel W Turner III and Nicole Hagstrom-Schmidt. Qualitative interview design. *Howdy or Hello? Technical and Professional Communication*, 2022.

[34] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. Selecting empirical methods for software engineering research. *Guide to advanced empirical software engineering*, pages 285–311, 2008.

[35] Richard E Boyatzis. *Transforming qualitative information: Thematic analysis and code development*. sage, 1998.

[36] Sharon Ryan and Rory V O'Connor. Acquiring and sharing tacit knowledge in software development teams: An empirical study. *Information and Software Technology*, 55(9):1614–1624, 2013.

[37] RK Kavitha and MS Irfan Ahmed. A knowledge management framework for agile software development teams. In *2011 International Conference on Process Automation, Control and Computing*, pages 1–5. IEEE, 2011.

[38] Sharon Ryan and Rory V O'Connor. Social interaction, team tacit knowledge and transactive memory: Empirical support for the agile approach. 2012.

[39] Reyes Juárez-Ramírez, Rafael Pimienta-Romo, and Violeta Ocegueda-Miramontes. Using social networks for integrating a tacit knowledge repository to support the performance of software development teams. In *International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making*, pages 167–179. Springer, 2013.

[40] Alberto Heredia, Javier Garcia-Guzman, Antonio Amescua, and Maria-Isabel Sanchez-Segura. Interactive knowledge asset management: Acquiring and disseminating tacit knowledge. *Journal of Information Science and Engineering*, 29(1):133–147, 2013.

[41] Thomas Chau and Frank Maurer. Knowledge sharing in agile software teams. In *Logic versus approximation*, pages 173–183. Springer, 2004.

[42] José Jairo Camacho, JM Sanches-Torres, and Ernesto Galvis-Lista. Understanding the process of knowledge transfer in software engineering: a systematic literature review. In *The International Journal of Soft Computing and Software Engineering [JSCSE]. Special Issue: The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13], Doi: 10.7321/jscse. v3. n3. 33 e-ISSN: 2251*, volume 7545, pages 219–229, 2013.

[43] Shahla Ghobadi. What drives knowledge sharing in software development teams: A literature review and classification framework. *Information & Management*, 52(1):82–97, 2015.

[44] Mansooreh Zahedi, Mojtaba Shahin, and Muhammad Ali Babar. A systematic review of knowledge sharing challenges and practices in global software development. *International Journal of Information Management*, 36(6):995–1019, 2016.

[45] Ted Foos, Gary Schum, and Sandra Rothenberg. Tacit knowledge transfer and the knowledge disconnect. *Journal of knowledge management*, 2006.

[46] Thomas Chau, Frank Maurer, and Grigori Melnik. Knowledge sharing: Agile methods vs. tayloristic methods. In *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.*, pages 302–307. IEEE, 2003.

[47] Kaj U Koskinen, Pekka Pihlanto, and Hannu Vanharanta. Tacit knowledge acquisition and sharing in a project work context. *International journal of project management*, 21(4):281–290, 2003.

[48] Deon Takpuie and Maureen Tanner. Investigating the characteristics needed by scrum team members to successfully transfer tacit knowledge during agile software projects. *Electronic journal of information systems evaluation*, 19(1):pp36–54, 2016.

[49] Lise Heeager and Peter Axel Nielsen. Agile software development and the barriers to transfer of knowledge: an interpretive case study. In *Scandinavian conference on information systems*, pages 18–39. Springer, 2013.

[50] Hanna Dreyer and Martin G Wynn. Tacit and explicit knowledge in software development projects: A combined model for analysis. *International Journal on Advances in Software*, 9(3/4):154–166, 2016.

[51] Martin Spraggon and Virginia Bodolica. Collective tacit knowledge generation through play: Integrating socially distributed cognition and transactive memory systems. *Management Decision*, 2017.

[52] Emmanuel Samuel Mtsweni and Nehemiah Maveterra. Issues affecting application of tacit knowledge within software development project. *Procedia computer science*, 138:843–850, 2018.

[53] Delio Ignacio Castaneda and Paul Toulson. Is it possible to share tacit knowledge using information and communication technology tools? *Global Knowledge, Memory and Communication*, 2021.

[54] Hariaty Mohd Idrus et al. Tacit knowledge in software testing: A systematic review. In *2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS)*, pages 1–6. IEEE, 2019.

[55] Hanna Dreyer. *Tacit Knowledge in a Software Development Project*. PhD thesis, University of Gloucestershire, 2018.

[56] Marie Liliane Isabelle Gervigny and Soulakshmee D Nagowah. Knowledge sharing for agile distributed teams: A case study of mauritius. In *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS)*, pages 413–419. IEEE, 2017.

[57] Emmanuel Samuel Mtsweni and Nehemiah Mavetera. Soft issues that limit sharing of tacit knowledge within software development project teams. In *2019 IEEE AFRICON*, pages 1–6. IEEE, 2019.

[58] Iris Buunk, Colin F Smith, and Hazel Hall. Tacit knowledge sharing in online environments: Locating 'ba'within a platform for public sector professionals. *Journal of Librarianship and Information Science*, 51(4):1134–1145, 2019.

[59] Maylon F De Brito, Rejane Maria Da Costa Figueiredo, Elaine Venson, Edna Dias Canedo, and Luiz Carlos M Ribeiro Junior. Knowledge transfer in a management process for outsourced agile software development. In *Proceedings of the 50th Hawaii International Conference on System Sciences*, 2017.

[60] Xin Zhou, Yuqin Jin, He Zhang, Shanshan Li, and Xin Huang. A map of threats to validity of systematic literature reviews in software engineering. In *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)*, pages 153–160. IEEE, 2016.

Appendix A

Invitation Letter

A.0.1 Intvitation letter to participate in Interview

Greetings <name of the practitioner>,

We are doing our thesis in software engineering on the topic "Tacit knowledge transfer in agile software development" at the Blekinge Institute of Technology under the supervision of KRZYSZTOF WNUK.

This questionnaire is designed to find the mechanisms used to transfer tacit knowledge, the challenges faced during the transfer of tacit knowledge, and the mitigation strategies to overcome the difficulties faced during tacit knowledge. We want to invite you to a 40 – 60 minutes interview to get a better hands-on understanding of this.

We would appreciate it if you took the time for our interview and shared your knowledge and experience. The interview would greatly help with understanding the transfer of tacit knowledge better. If you are interested in the discussion, don't hesitate to get in touch with us. We are looking forward to hearing from you.

Best Regards,
M.Prerana Rao
G.Preetham Raj

Master Thesis Supervisor:
KRZYSZTOF WNUK.
Department of Software Engineering.

Appendix B

Interview Questionnaire

1. Job Title
2. Number of Years of experience.
3. Do you have experience working in agile software development? Any other methodologies have you worked on?
4. Have you ever participated in knowledge transfer?
5. Which type of agile team have you worked in?
6. What mechanism did you or the organization is using currently to transfer tacit knowledge transfer? a. Any different methods for a different types of teams?
7. What are the main challenges you have faced while doing tacit knowledge transfer?
8. What strategies you have followed to overcome the challenges for better tacit knowledge transfer?
9. What do you think is the most effective mechanism based on your experience?

Appendix C

Screenshots of Data Analysis

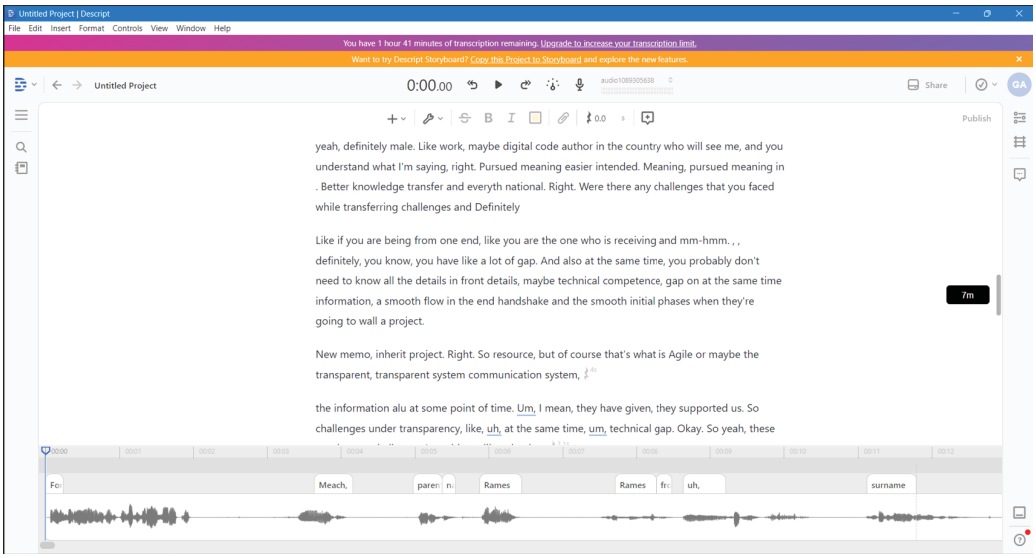

Figure C.1: Represent Transcribing

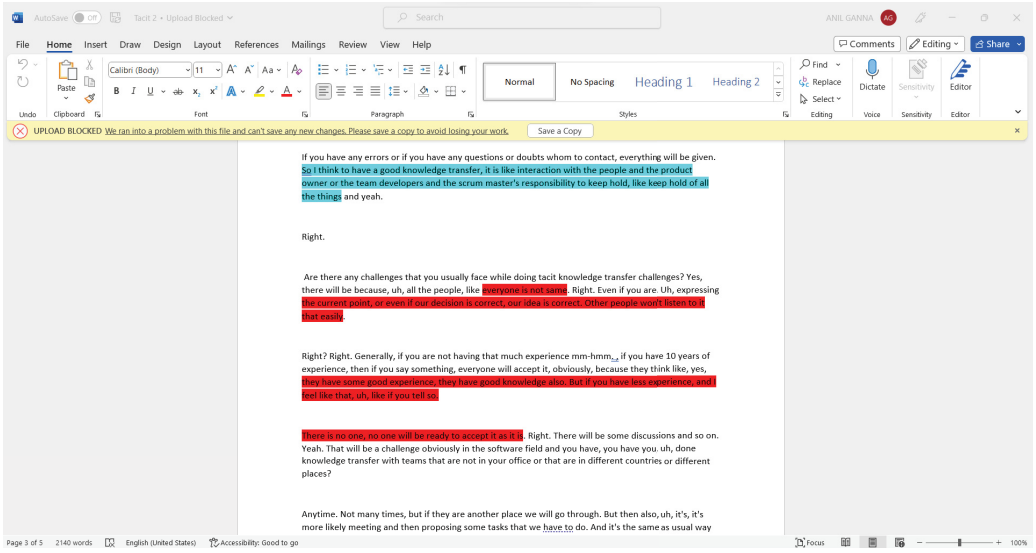

Figure C.2: Color coding of Initial code from Interviews

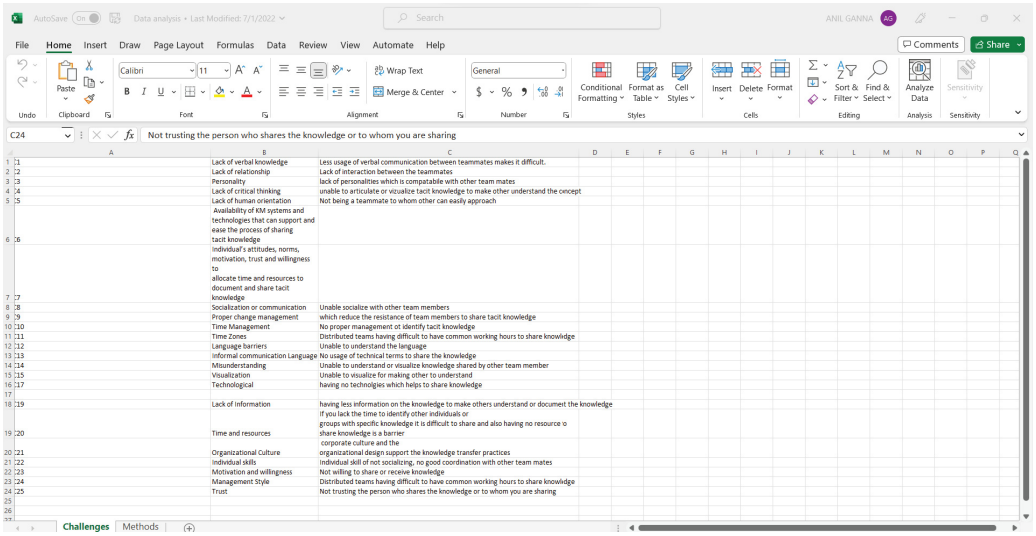

Figure C.3: Represent Interview data analysis

Appendix D

Quality Assessment Scores

Quality Assesment							
Study ID	Questions Score						Total Score
	C1	C2	C3	C4	C5	C6	
P1	1	0.5	0.5	0	1	0.5	3.5
P2	0.5	1	0	1	0	0.5	3
P3	1	0	0	0.5	0.5	1	3
P4	1	1	0.5	1	1	0	4.5
P5	0.5	0	1	0.5	0.5	1	3.5
P6	0	0.5	0	1	1	0.5	3
P7	1	0	1	1	0	0.5	3.5
P8	1	1	1	0	1	1	5
P9	0.5	0.5	1	1	1	1	5
P10	0	0	1	1	0.5	1	3.5
P11	0	0.5	1	1	0.5	0.5	3.5
P12	0.5	0	1	1	0	0.5	3
P13	1	0.5	0.5	0.5	0	1	3.5
P14	0.5	1	1	0.5	0.5	1	4.5
P15	1	1	1	1	0.5	1	5.5
P16	0.5	0.5	1	1	1	0	4
P17	0.5	1	0	0	1	1	3.5
P18	0	0.5	0.5	0.5	1	1	3.5
P19	0.5	0.5	1	1	1	0	4
P20	0.5	1	0	0.5	0	1	3
P21	0.5	1	0.5	0	1	0.5	3.5
P22	0	0	0	1	1	1	3
P23	0.5	1	0.5	0	0	1	3
P24	1	1	1	0.5	1	1	5.5
P25	0.5	0.5	1	0	1	0	3
P26	1	0.5	1	0	0.5	0	3
P27	0.5	0.5	0	1	0.5	0.5	3
P28	0.5	1	1	0	0.5	0.5	3.5
P29	1	0.5	1	1	1	0.5	5
P30	1	0	1	0.5	0	1	3.5
P31	1	0	1	0.5	1	0	3.5

Figure D.1: Scores of Quality Assessment

Faculty of Faculty, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden