Knowledge development and knowledge sharing in expert communities

A qualitative study on knowledge development and knowledge sharing in expert communities facilitated by working with joint epistemic objects

Eirik Kaleva Turunen Rise and Maja Greni Sundland

Master thesis in Pedagogy

Knowledge development and work life learning

45 credits

Institute of Pedagogy

The Faculty of Educational Sciences

UNIVERSITY OF OSLO

Spring 2022

“Objects of knowledge appear to have the capacity to unfold indefinitely”

- Karin Knorr Cetina (2001)

SUMMARY

MASTER THESIS IN PEDAGOGY – KNOWLEDGE DEVELOPMENT AND WORK LIFE LEARNING

Title	Knowledge development and knowledge sharing in expert communities – A qualitative study on knowledge development and knowledge sharing in expert communities facilitated by working with joint epistemic objects
Authors	Eirik Kaleva Turunen Rise and Maja Greni Sundland
Subject	PED4491
Semester	SPRING 2022

Keywords
• Knowledge sharing • Knowledge development • Expert communities • Epistemic objects • Epistemic practices • Software development • Computer simulation • Data modelling • Acoustical engineering

Summary

Theme and problem area

In today's society expert knowledge has earned a high status where work is increasingly specialised and knowledge intensive (Knorr Cetina, 2001). Solving complex problems requires an understanding of how knowledge is being developed and shared by the people who engage in highly specialised knowledge work. In this study, we explore how knowledge development and knowledge sharing takes place in an expert community that develops software for room acoustics through investigating the role of epistemic objects and practices and the challenges the expert community faces when developing the software.

We have explored how a team of experts in acoustic engineering organise their joint development of complex software by asking the broader research question: *How is the development and sharing of knowledge in expert communities facilitated by working with joint epistemic objects?*

The research problem is answered by examining the following research questions:

- *In what ways does this expert community develop knowledge by way of epistemic objects and epistemic practices?*
- *How does knowledge sharing take place related to their work on epistemic objects?*
- *What challenges does this expert community face when working on joint epistemic objects?*

The theoretical framework that has been utilised to investigate the research questions is based on the concepts of *expert communities* (Knorr Cetina, 1999, 2001), and its special place of knowledge in contemporary society. *Epistemic objects* (Knorr Cetina, 1997, 2001), dominate much of the expert community’s work, and in our study represents the shared epistemic object developed by the expert community. In this thesis we also explore the role of *knowledge sharing* activities in the practices of developing epistemic objects in expert communities. And finally, *epistemic practices* (Knorr Cetina, 2001) that concerns specific practices and dynamic relationship experts engage in when developing knowledge and practice, or what experts do when working with epistemic objects. The purpose of this theoretical framework is to shed light on how knowledge is developed and shared in an expert community that works with joint epistemic objects.

Method and data collection

To study how knowledge is developed and shared in expert communities when working with joint epistemic objects, we decided on a qualitative approach where we conducted a qualitative observation- and interview study. The data material was collected through both digital and physical semi-structured interviews and observations of a team of experts that collaborated on developing a software for room acoustic simulation. The recruitment process was based on a strategic selection, and the participants were of empirical interest as they were a team of highly specialised knowledge workers collaborating on developing a complex object. The collected data were analysed through thematic analysis.

Results and conclusion

We approached a team of experts forming an expert community around a room acoustic simulation software they collaborated on developing. As specialists in acoustic engineering their work was embedded in a wider knowledge culture within their field, and they participated in specific knowledge cultures for developing the software. The specialists engaged with the software as an epistemic object to solve complex acoustical knowledge problems in unfolding processes of exploration and stabilisation, applying scientific knowledge to concrete cases. Working with epistemic objects required knowledge sharing to coordinate the variegated knowledge of the expert community. Specific functions in the software acted as focus points for software development and contributed to developing the wider field of expertise. By engaging in epistemic practices of engineering they shared knowledge and solved complex knowledge problems linked to wider machineries of knowledge construction through practices of engineering in social contexts, uses of data and evidence to make decisions, tools and strategies for problem solving and finding solutions through creativity and innovation. The expert community was faced with challenges of complexity and uncertainty originating from working on highly complex epistemic objects and employed social strategies of communication and collaboration to handle these challenges. Companies developing epistemic objects need to encourage frequent interactions and communication to facilitate for expert communities to develop their own strategies to cope with the challenges embedded in working with epistemic objects.

Preface

After five years studying pedagogy, we are at the end of a learning journey where we have been introduced to knowledge and perspectives, we never could not have foreseen, and that have shaped us as both students and humans in ways we still don't fully understand.

The journey culminated with this thesis, our own little epistemic object that came to be through grappling with knowledge problems and trying to find answers to pedagogical questions of knowledge development. We would not have succeeded without the case company in this thesis that warmly welcomed us and let us have a glimpse of their world. Our utmost thanks to our team of experts and the gatekeepers who opened the doors, it was not just a learning experience, but a lot of fun to observe your community.

A special thanks to our supervisor whose deep understanding of the subject seems to have no end, and whose feedback has been invaluable and contributed to a deeper understanding and respect of academic work.

Finally, thanks to each of the authors of this thesis. When we first met on the first day of university, we did not know we would end up here, but here we are in the final days of our master's degree, and learning together has made what must be the toughest challenge in our academic life both fun and meaningful.

Maja

I would like to send a big thank you to my boyfriend and my best little daughter for all the love, care, and patience you have shown during this process. You have supported me, cheered me on, and your quote "mamma jobber SÅ hard!" have motivated me all the way. I am so grateful. Thank you to my fellow students who have made my five years at Helga Eng's a pleasure to look back on. Now, I look forward to a new chapter with new opportunities and what looks like an exciting future.

Eirik

Firstly, I would like to send a special thanks to my partner, whose patience and understanding has been endless, as well as my family and friends. Thank you for all your support. I must thank the staff of researchers and administrators at the University of Oslo who have given

time, dedication, humour, enthusiasm, support and genuine concern throughout our studies and helped us patiently. Thank you for all I have learned and experienced together with my fellow students. Having finished after five years of work and studies combined, I look forward to slowing down and applying newly developed knowledge into my own practice and have time to digest what it all will mean for my future work life.

Work distribution

In line with formal requirements, we will here elaborate on how we divided the work between us due to the fact that we have been collaborating on implementing and writing this master thesis. We collaborated closely and maintained a systematic approach throughout the entire project. Nevertheless, it was necessary due to the extent of the study to distribute the largest areas of responsibility between us, respectively the *review, theory and methodology* chapters of the study. We prepared these chapters individually before coordinating them with each other. The remaining chapters such as *introduction, analysis, discussion and conclusion* have mostly been developed in collaboration. We preferred to use Google Docs as our work tool to write this thesis which contributed to maintaining a transparent work process. We were both equally involved in the preparation of the interview- and observations guides, and also in the process of conducting the interviews and observations where we shared the responsibility of conducting and later transcribing the interviews equally between us. Throughout the process we have both been present and participating during all phases of the project and we share equal ownership of this study.

Oslo, June 2022

Content

1. Introduction 1
1.1 Presentation of theme and relevance 1
1.2 Research problem and research questions 3
1.3 Theoretic framework and demarcation 3
1.4 Empirical context and methodology 5
1.5 Thesis structure 5
2. Review 7
2.1 Introduction 7
2.2 Software development 8
2.2.1 Agile software development practices 8
2.2.2 Continuous software development practices 9
2.2.3 Evolutionary software development practice 10
2.2.4 Knowledge sharing and software development practices 12
2.3 Computer simulations as epistemic objects 14
2.3.1 Computer simulation's role in epistemic innovation 15
2.3.2 Computer simulation of room acoustics 16
2.4 Summary 17
3. Theory 19
3.1 Characteristics of expert communities 19
3.2 Expert communities as object-centred 20
3.3 Knowledge sharing through object construction 22
3.4 Working with epistemic objects 23
3.5 Epistemic practices 26
3.5.1 Epistemic practices in engineering 27
3.6 Summary 29

4. Methodology 32
4.1 Research Design 32
4.2 Selection of case and recruitment of participants 35
4.3 Approaches to data collection 36
4.4 Analytical Approach 42
4.5 Ethical considerations 49
4.6 Credibility / The quality of the study 50
5. Analysis 52
5.1 Introduction 52
5.1.1 The case company 52
5.2 Working as an expert community 54
5.2.1 The expert community and object work 56
5.2.2 The expert community and epistemic objects 57
5.2.3 The Omni source and the material calculator 62
5.2.4 The Omni source as an epistemic object 63
5.2.5 The material calculator as an epistemic object 66
5.3 Epistemic practices 70
5.3.1 Epistemic practices involved in developing the Omni source 73
5.3.2 Epistemic practices involved in developing the material calculator 75
5.4 Knowledge sharing through working on objects 77
5.4.1 Knowledge sharing and the Omni source 88
5.4.2 Knowledge sharing and the material calculator 90
5.5 Challenges 92
5.6 Summary 94
6. Discussion 96
6.1 Introduction 96

6.2 In what ways does this expert community develop knowledge by way of epistemic objects and epistemic practices? 96
6.2.1 Expert knowledge development through epistemic objects 97
6.2.2 Expert knowledge development through epistemic practices 102
6.3 How does knowledge sharing take place related to their work on epistemic objects? 105
6.4 What challenges does this expert community face when working on joint epistemic objects? 108
6.5 How is the development and sharing of knowledge in expert communities facilitated by working with joint epistemic objects? 110
7. Conclusion 113
7.1 Summary 113
7.2 Implications 113
7.3 Future research 114
7.4 Limitations 115
8. Literature 117
Appendices 121
A. Privacy Representative Approval from Norwegian Centre for Research Data 121
B. Info letter 123
C. Consent form 126
D. Interview guide - individual interviews 127
E. Interview guide - follow-up interview 129
F. Interview guide - group interview 130
G. Observation form (in Norwegian) 131

Tables

Table 1: Epistemic practices in engineering 28
Table 2: Characteristics of epistemic objects..30
Table 3: Data collection 35
Table 4: Thematic analysis 45
Table 5: Categories of knowledge sharing activities 78
Table 6: Epistemic practices of engineering 103

Pictures

Picture 1: Initial codes 46
Picture 2: Searching for themes 47
Picture 3: Reviewing themes 48
Picture 4: Audio visual immersion lab of the university 52
Picture 5: The anechoic chamber of the university.. 53
Picture 6: The team working in the shared office space 53
Picture 7: Sketch of the Omni source 64
Picture 8: 3D-printed prototypes for the Omni source.. 65
Picture 9: Material calculator user interface 67
Picture 10: Presentation of the specially designed suitcase for the Omni source 74
Picture 11: Documentation - development of a theoretical paper on the Omni source. 80
Picture 12: The teams weekly meeting 83
Picture 13: Feedback from the team on the Omni source 85
Picture 14: Collaboration and problem solving, discussing graphics for the scientific paper on the Omni-source 87

1. Introduction

1.1 Presentation of theme and relevance

The growing importance of knowledge and its special place in contemporary society has been recognised by the increasing trend of work tasks that require expert knowledge (Knorr Cetina, 1999, 2001), and the term knowledge society has been used to describe the current realities. Expert knowledge is highly specialised, it is shared in *expert communities* and developed over a longer period of time. This type of knowledge has received a high status in today's society as more work is increasingly specialised and knowledge intensive (Knorr Cetina, 2001). To understand how the increasing focus on knowledge intensive work is affecting society it is necessary to understand the practices experts use when developing and sharing knowledge in highly specialised knowledge work. Expert communities form around complex problems linked to specific areas of expertise, and they take part in processes of learning and knowledge development linked to wider knowledge cultures specific for their expert area (Knorr Cetina & Reichmann, 2015).

In this master's thesis we will investigate how knowledge development and knowledge sharing takes place in a particular environment within the field of acoustic engineering. This kind of environment is often referred to as *expert communities* (Knorr Cetina, 1999, 2001). Typical for an expert community is that it often works with what we refer to in this thesis as *epistemic objects* (Knorr Cetina, 1997, 2001), where the object is in the centre of the knowledge development. The epistemic object is the point of focus in a process of exploration where experts try to solve the complex knowledge problem it raises (Knorr Cetina, 2001).

The use of expert knowledge to resolve complex problems is a characteristic of professional experts and experts are also producers of knowledge through participation in wider fields of expertise (Knorr Cetina & Reichmann, 2015). As professions, such as engineering, have evolved and become more interdisciplinary and interventionist, knowledge is more distributed and more diverse groups of actors engage in knowledge development. This means that more knowledge is being developed through the combination of research and practical application through expert communities that are part of expert systems linking the wider areas of expertise (Knorr Cetina & Reichmann, 2015).

"Epistemic practices are the socially organised and interactionally accomplished ways that members of a group propose, communicate, justify, assess, and legitimise knowledge claims" (Cunningham & Kelly, 2017, p. 487). This is a general definition, in this thesis we focus on epistemic practices in relation to our broader research problem. Software development can be seen as an example of working with epistemic objects through epistemic practices, as exemplified by Knorr Cetina (2001) who describes software as an example of epistemic objects and programming as an example of epistemic practices.

Knowledge sharing is central to both collaboration and software development in teams to overcome challenges and bring knowledge development further (Ghobadi, 2015). More research is needed on how expert communities establish knowledge sharing cultures and how they affect the dynamics of knowledge development (Wang & Noe, 2010). Recognising knowledge sharing drivers in software development teams is beneficial for the members and helps them reach closer to ideal levels of knowledge sharing and communication. Less is however known about the technology related knowledge sharing drivers, including how methods are used, the role of objects in software development, and task-related drivers, such as task uncertainty, the lack of necessary information to develop the software (Ghobadi, 2015). To understand how engineers develop knowledge, these knowledge generating practices need to be studied based on the social processes of meaning making that take place in engineering expert communities (Cunningham & Kelly, 2017).

Engineering work consists of practices of knowledge sharing and development (Cunningham & Kelly, 2017) and needs to be observed in specific contexts such as acoustical engineers. At the same time such practices will necessarily take different forms in specific types of engineering.

In our thesis we contribute to the field by exploring how a team of experts in acoustic engineering organise their joint development of complex software. Our aim is to bring new insight into how knowledge development and knowledge sharing take place in an expert community developing a software for room acoustics through examining the role of epistemic objects and practices and the challenges they face.

1.2 Research problem and research questions

The aim of this master thesis is to explore how knowledge sharing and development takes place in highly specialised knowledge work. Our thesis addresses the following broader research problem:

How is the development and sharing of knowledge in expert communities facilitated by working with joint epistemic objects?

To be able to provide an adequate answer to our research problem, we have studied a team of eight experts that collaborate on developing a room acoustic software and based our interviews and observations on the following research question:

- *In what ways does this expert community develop knowledge by way of epistemic objects and epistemic practices?*
- *How does knowledge sharing take place related to their work on epistemic objects?*
- *What challenges does this expert community face when working on joint epistemic objects?*

1.3 Theoretic framework and demarcation

We have delimited our study presented in this thesis by applying a theoretical framework based on the concepts of *expert communities, epistemic objects, epistemic practices and knowledge sharing.*

Knorr Cetina (Knorr Cetina, 1999, 2001) introduced the concept of *expert communities,* and its relation to the special place of knowledge in contemporary society. Expert communities are groups of specialised professionals who form around complex problems linked to specific areas of expertise (Knorr Cetina & Reichmann, 2015) and engage in specific knowledge practices to solve these problems, manage and coordinate their work (Nerland & Jensen, 2012). We consider this a relevant concept in our study as the team we have chosen as our case consists of experts within the fields of acoustical engineering and software development who work together on developing a shared epistemic object. Experts are highly object-centred, and objects dominate much of their work (Knorr Cetina, 1997, 2001). Working on the objects is a driving force for the expert communities to develop their understanding of the problem and the field of expertise itself (Werle & Seidl, 2015, p. 70). In our study the shared epistemic objects relate to the room acoustic software that the expert community collaborate on developing.

Expert communities develop epistemic objects in specific ways through generating and sharing knowledge (Nerland & Hasu, 2021, p. 66). In this thesis we will explore the role of knowledge sharing in the practices of developing epistemic objects in expert communities. By establishing relationships with their epistemic objects and engaging in knowledge processes, expert communities can identify with their expert field leading to a higher level of engagement and motivation. Epistemic objects motivate experts to mobilise knowledge flows and increase the continuity in knowledge generation. This motivation comes from an urge to know more, and keep the experts together in joint knowledge work formed around the epistemic object through shared interest, common goal and a need to know what the others know (Liu, 2019).

Epistemic practices are the specific practices and dynamic relationship experts engage in when developing knowledge and practice, or what experts do when working with epistemic objects. Epistemic practices enable experts to explore complex problems, access knowledge resources and identify open opportunities and temporal solutions (Knorr Cetina, 2001; Nerland & Jensen, 2012). Expert communities engage in epistemic practices when working with epistemic objects, and these practices take place when an expert is faced with a new problem which makes them have to relate to an object in a different way (Knorr Cetina, 1997, 2001). When developing the epistemic objects, the team of experts engage in software development practices. Cunningham and Kelly (2017) have identified significant epistemic practices in engineering and categorised these in relation to four areas of practice: *social contexts, using data and evidence to make decisions, tools and strategies for problem solving, and finding solutions through creativity and innovation*.

In chapter three we develop these concepts and perspectives further and present an analytical framework that will be used in the analysis to inform categories and structure the presentation of our data, demonstrating how expert communities share and develop knowledge by working with epistemic objects, and then to discuss how our findings relate to and inform the theoretical concepts in the discussion.

1.4 Empirical context and methodology

To address the overall research problem and answer research questions of how experts develop and share knowledge in expert communities when working with epistemic objects we have focused the empirical work on a company run by a small team of experts collaborating on developing a highly specialised room acoustic computer simulation software. The software has been developed for over 30 years and the team of expert developers represents a complex field of expertise involving both programming competence, scientific knowledge and expertise of acoustical engineering. To understand how the knowledge development takes place we employed qualitative research methods as identified as most appropriate to investigate constitutive processes of some phenomenon (Silverman, 2017). We have observed the team over a period of seven weeks where we attended some of their meetings and conducted semi-structured interviews both digitally and physically. The observations and interviews gave us the opportunity to explore the ways the team of experts worked when collaborating on developing the software, how they shared and worked with knowledge and what challenges they faced in this process.

1.5 Thesis structure

The thesis is organised in 7 main chapters:

We start by introducing the theme and relevancy for our study by presenting our main research problem and associated research questions. This first chapter provides an overview of our theoretical framework and demarcation, and empirical context and methodology, as well as an overview of the thesis structure.

Chapter two presents review literature and relevant research on software development as well as insight on knowledge sharing within software development. The chapter also presents relevant research on computer simulation as an epistemic object and computer simulation of room acoustics.

The third chapter presents a theoretical framework which the thesis is based on. The framework is intended to function as a tool for analysing our data material and will support further discussions and reflections that will contribute in providing answers to the research problem.

In the fourth chapter we present our methodological decisions and considerations. We review key aspects related to our study's research design, the collection and data processing and ethical considerations protecting the participant's autonomy. Finally, we present reflections on the credibility of our study.

In chapter five we present our analysis based on our collected material. The analysis is intended to provide a relevant base for the thesis discussion and contribute to answering the research problem and associated research questions.

The discussion of the thesis is presented in chapter six. The discussion is based on our three research questions, and we argue these in the light of the thesis' theoretical framework and collected data and then relate our findings to the broader research problem.

Finally, in chapter seven we conclude based on our findings and present our contributions, proposals for further research and limitations.

2. Review

2.1 Introduction

This thesis raises the issue of how expert communities develop and share knowledge by working with epistemic objects. Epistemic objects can take many shapes, in the case elaborated in this thesis the epistemic objects take the form of a highly complex computer software, more specifically advanced room acoustics computer simulation software. Research has explored software development, identified specific trends and established methodologies (Dingsøyr et al., 2012; Fitzgerald & Stol, 2017; Ghobadi, 2015) that in varying degrees are followed (Dittrich et al., 2020, p. 2). In particular we have identified agile, continuous and evolutionary software development as relevant for this thesis, and we have found that knowledge sharing is a critical part of the development of computer software (Ghobadi, 2015). In object construction, computer models have been studied as epistemic objects and research has been conducted on computer simulations, demonstrating the value of epistemic theories when trying to understand what is going on when computer simulations are used to develop knowledge (Merz, 2018, p. 335).

Computer simulation of room acoustics in particular is not a new phenomenon, but can look back at its 60 years history (Vorländer, 2013). With the rise of new and more powerful personal computers, the use of computer simulation in room acoustics became more widespread (Vorländer, 2011) and changed acoustics from numerous calculations into advanced virtual simulations that made it possible to observe results in practice. These advantages of advanced room acoustics computer simulation has made its use widespread in the consulting industry (Forsyth, 2018). Making room acoustic computer simulations is however not straightforward and research has identified the main technical challenges in making room acoustics computer simulation work in practice (Vorländer, 2013). Tests made comparing the most well-known software on the market showed this type of software's weaknesses and lacks when trying to simulate how sound would behave in not-yet existing realities (Brinkmann et al., 2019).

Even though research has explored the challenges in the technology related to acoustics computer simulation software, limited research could be found on how the expert communities behind the software share and develop knowledge when working with the computer software. This case contributes to the field of research by giving an example of how this challenging and uncertain endeavour can take place.

In this chapter we first review some of the more influential methodologies in software development over the last 20 years, before reviewing how computer models and simulation in general, and computer simulation of room acoustics in particular, is conceptualised in some of the research available today.

2.2 Software development

Software development practices have been influenced by changing trends and methodologies, some of which have received more attention than others (Dingsøyr et al., 2012; Dittrich et al., 2020; Fitzgerald & Stol, 2017; Ghobadi, 2015). We will here briefly present some of the more influential, namely agile software development (Dingsøyr et al., 2012; Ghobadi, 2015; Kumar & Bhatia, 2012) continuous software development (Fitzgerald & Stol, 2017) and practice oriented software development (Dittrich et al., 2020). Knowledge sharing also holds a central position in software development, which can be described as a collaborative and knowledge intensive process (Ghobadi, 2015).

2.2.1 Agile software development practices

Agile methodology started with the publishing of a manifesto in 2001 which are claimed to bring unprecedented changes to the software development field (Dingsøyr et al., 2012). Where traditional methods were not seen as efficient enough to adapt to rapid changes in software demands, the efficiency of agile methods transformed the software industry, seeing software development as an iterative and cognitive process based on incremental development (Kumar & Bhatia, 2012). The manifesto marked a shift in how software was developed, giving spark to a wide range of working methods such as extreme programming, scrum, lean software development, feature-driven software development and crystal methodology which all aim to address the core principles of the manifesto (Dingsøyr et al., 2012). The core four characteristics of the *agile manifesto* is that it values individuals and interactions rather than processes and tools, a working software over comprehensive documentation, collaboration with the customer rather than negotiating contracts, and responding to changes over following plans. Even if every mentioned aspect has value, in the agile manifesto some are deemed more important than others (Dingsøyr et al., 2012, p. 1).

Agile methodology has taken software development a big step towards collaborative development, where people could themselves decide over the processes that formerly constrained them (Dingsøyr et al., 2012). According to Kumar & Bhatia (2012), the main benefits of agile methodology are higher quality software in a shorter time, self-organising teams and successful customer collaboration. Implementation of agile methodology involves formal face-to-face communication between team members where they discuss what they are working on and inform the others on their progress and challenges. The methodology is deemed favourable due to improved communication, quick releases and flexibility of the design process. Limitations on the other side, are high requirements of coordination and communication from project managers and the focus on development at the expense of design and use (Kumar & Bhatia, 2012).

Dingsøyr et. al (2012) reviews the research on agile methodology. By examining publications and citations they outline the field and explain how the method has developed. Research varied from how to adopt the method and its efficiency, to different sides of team dynamics such as trust, self-organising and communication. It has also investigated the consequences of test-driven development and issues of implementation in distributed work environments. The study contributes to developing the understanding of the implications of agile software development, while asking for more efforts to be invested in further investigating its fundamental principles, and Dingsøyr et al. (2012) finds an urgent need for more studies on mature agile development teams as most projects were focusing on new users. Further, research is needed on the adoption of agile to specific projects and a more theory rooted examination of the various practices, to strengthen the theoretical framework. Hence, better understanding of how collaborative software development is facilitated is needed.

2.2.2 Continuous software development practices

Continuous software development is an umbrella term for different initiatives that can be termed *continuous*. The rationale builds on the increasing need for a continuous end-to-end flow between customer demand and fast service delivery. Software development has suffered from a disconnection between the central activities of planning, development and implementation, making activities such as planning, testing, integration and releases implemented in a casual and sporadic way. By increasing the frequency of certain critical activities and eliminating the discontinuities between development and deployment many of these challenges can be prevented. Releasing new versions often and early can give benefits when it comes to the quality and consistency of software development. Errors can be detected and fixed sooner when the distance between development and execution is shorter (Fitzgerald & Stol, 2017). In their research agenda for continuous software engineering, Fitzgerald & Stol (2017) sets the direction for future research, asking amongst others how continuous evolution and maintenance of software systems can be facilitated, how hardware and software can be co-developed using a continuous software engineering approach and how key barriers between development and operations can be removed.

2.2.3 Evolutionary software development practice

Dittrich et al. (2020) looks into how software development processes develop over time, a relevant aspect for the software we later explore, which has a history of being developed continuously for over 30 years. They find little research on how situated software development practices develop, and the observations that exist don't show if the practices are results of conscious and deliberate processes or unintended decay. Further, they find largely no research on the relationship between descriptions of methods and the practices they are to inform and improve. While principle 12 of the agile manifesto states that “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.” (*Manifesto for Agile Software Development*, n.d.), Dittrich et al. (2020) claims that little research exists that describes what happens at these reflection meetings and how practices adapt accordingly.

According to Dittrich et al. (2020), when a software is developed over a longer time software development practices tend to drift away from structured and agile methods. They ask the questions of how teams coordinate joint software development, and if practice degenerates or evolves based on conscious deliberation and reflection. Dittrich et al. (2020) see software design and development practices through the perspective of Knorr Cetina as an unfolding practice that evolves as software is being developed. Based on the ethnographic study of a small and successful team of software service providers, Dittrich et al. (2020) finds a bottom-up software development practice resulting from conscious deliberation where the team was a natural part of the software development practice, as described by the 12th principle of the agile manifesto. The team engaged in experimentation as a way to explore new ways of working and kept up and developed joint practices that met their changing and evolving needs.

While we found the research on epistemic practices in software development to be relatively limited, one study explored how software development teams in a medium sized company use routine in resolving and dealing with highly complex and ambiguous issues (Mahringer et al., 2019). The authors looked at how software is developed by exploring how multiple routine performances helped facilitate and organise open-ended, epistemic processes in software development by structuring, materialising, recalling and closing the processes. Work in organisations is often characterised by high degrees of complexity, uncertainty and ambiguity, in these cases the outcomes of routine can only partially be predefined and constantly changed over time (Mahringer et al., 2019). In the case, routine performances materialised (created new knowledge) and recalled (introduced existing knowledge) partial epistemic objects that revealed new lacks and needs, leading to routines being broken and actions of reperforming, repeating, anticipating and skipping taking place.

Performing routine helped restrict the unfolding and open-ended process, giving it a sense of order and protecting the overall software development activities from risk. Balancing the need of order and wish to explore created tensions demonstrated by emotional reactions of the experts towards both the routine and the unfolding activities. By using positive emotions to counteract the negative emotions they could achieve emotional balance. This didn’t make open-ended processes more predictable, but facilitated innovation by orchestrating the process as it unfolded in action, making space for unfolding and at the same time supporting coordination and making sense of what to do next (Mahringer et al., 2019). Nerland & Jensen (2014) suggest further research on knowledge practices in expert communities to focus on their ways of dealing with complexity and uncertainty.

2.2.4 Knowledge sharing and software development practices

Knowledge sharing is a knowledge-centred activity where employees contribute to the application and innovation of knowledge (Wang & Noe, 2010) or "the ease with which knowledge is transferred and deployed within a network" (Liu, 2019, p. 339). It involves providing task information and know-how to help others, and to collaborate with others in order to solve problems, develop new ideas or implement specific policies or procedures. It can take place through communication in writing or face-to-face, through networking with other experts, and by documenting, organising and capturing knowledge for the use of others. How knowledge is developed on a team and organisational level is affected by the degree of knowledge sharing taking place between employees (Wang & Noe, 2010, p. 117).

Wang & Noe (2010) reviews a wide range of literature from 1999 to 2008 to develop an organising framework and discuss issues, research needs and practical implications of knowledge sharing. Their framework covers organisational context, cultural, interpersonal and team characteristics, as well as motivational factors, individual characteristics, knowledge sharing perceptions and behaviours. While segmented structures are likely to inhibit knowledge sharing across functions and practice communities, knowledge sharing can be facilitated through decentralised organisational structure and a work environment that encourages interactions among employees. Facilitation practices can be the use of open workspace and fluid job descriptions, as well as encouraging cross-department communication and informal meetings. Further, organisations should create opportunities for employee interactions and deemphasize hierarchy. When it comes to if team characteristics and processes support knowledge sharing there were few studies, but they suggest that knowledge sharing is more likely if the team has been formed for a long time and experience high cohesiveness. Agreeable and extraverted communication styles were positively associated with knowledge sharing willingness and behaviours, the same with empowering leadership. Future research directions point towards the need for further investigation on how culture and norms in professional communities of practice are established, in order to understand how a knowledge sharing culture can be promoted and how it affects the dynamics of knowledge sharing and learning among employees and teams (Wang & Noe, 2010).

Knowledge sharing is a concern both in collaborative teams and software development teams (Ghobadi, 2015). Software development requires the mixing and merging of diverse knowledge that is spread around different fields of expertise. *Transactive memory* can be understood as the set of knowledge possessed by a group regarding an awareness of who knows what (Ghobadi, 2015, p. 87), helps drive knowledge transfer in distributed teams and can be achieved through frequent interactions between team members. Through developers' engagement in iterative development and quality validation cycles involving rapid reflections and frequent introspection across team members representing different specialisations, and through the exploitment of diverse expertise and the exploration of existing and potential opportunities in software development, highly needed intensive knowledge sharing can take place (Ghobadi, 2015).

Effective knowledge sharing is requisite for software development team members to discuss critical aspects of a project and overcome the cultural and social challenges that come from working across distributed spaces. It involves exchanging task-related information, ideas, know-hows and feedback regarding software products and processes while exploiting available resources, addressing perceived challenges, and exploring emerging opportunities in software development. Understanding what drives knowledge sharing in software development teams is needed for team members to monitor their own knowledge sharing patterns and programs, bringing them a step closer to ideal levels of communication and knowledge sharing (Ghobadi, 2015).

Ghobadi (2015) reviews the literature on knowledge sharing drivers in software teams from 1993-2011 in order to improve the understanding of knowledge sharing in software development teams. She integrates 44 drivers into a classification framework based on organisational change theory. The four main categories of the framework are *people-, structure-, task- and technology-related drivers*. Less research has been conducted on *technology-related drivers* where there are new research opportunities in project technology aspects such as methods-in-use, boundary objects and the role of standardisation in software development, and on *task-related drivers* where less attention has been paid to task uncertainty where lacking the information needed to develop the software can hinder effective knowledge sharing in distributed teams (Ghobadi, 2015). Categorisations of knowledge sharing drivers are important, at the same time it is important to view the processes and practices that take place in context and as more than individual factors.

Engineer knowledge work involves epistemic practices that contribute to knowledge sharing and development. In an article about the epistemic practices of engineering Cunningham &

Kelly (2017) reviews empirical studies of engineering in practice and categorises the epistemic practices of engineering in order to explain how knowledge is constructed through action and practice. They find that engineering is different from other disciplinary knowledge settings by its unique feature of being developed through practices where requirements are defined by clients and the relevant conditions of the situation where design work takes place. Epistemic practices help investigate, generate explanations for, and evaluate knowledge claims. Cunningham & Kelly (2017) base their perspective on the study of knowledge constructing communities found in science and technology studies, focusing on the social and interactional processes of reasoning that takes place in practices and actions taken in the world. Knowledge claims are developed and evaluated through coordinated and agreed upon efforts by social groups and then made legitimate through the weight of their evidence in social forums (Cunningham & Kelly, 2017).

Knowledge sharing is central to collaboration on teams. While some is known about the different drivers and benefits of knowledge sharing both in general and in engineering and software development teams in particular, it is important to understand how joint knowledge sharing happens in context as part of epistemic practices in software development by professional teams. In addition, computer simulations constitute a specific type of epistemic objects.

2.3 Computer simulations as epistemic objects

Computer simulations combine both aspects of software development and epistemic objects. Knorr Cetina (2001) uses computer software as an example of epistemic objects, and computer simulations are used in scientific exploration as an unique epistemic object (Merz, 2018). Outside the realm of science, experts work on epistemic objects to develop knowledge, but little research could be identified on computer simulations as epistemic objects. One article however, describes how scrum teams use routine to organise developing software as an epistemic object (Mahringer et al., 2019). We will here review literature on computer simulation software as epistemic objects, and the characteristics and challenges of room acoustics computer simulation software.

2.3.1 Computer simulation's role in epistemic innovation

Computer simulations have gained much attention in the field of science and technology as an epistemic practice that follows its own dynamic of innovation (Merz, 2018). The epistemic significance of computer simulations is shown in different ways in studies of science with a focus on how effective the models are in representing its research subject. Focusing on computer simulations as an epistemic object makes us aware of their practical use as well as their function as an object in themselves, and adding a practice perspective helps identify how computer models have been used in specific scientific contexts, their locations and roles. Computer models create implicit knowledge in their descriptive role as *models of* and they create explicit knowledge in their performative, instrumental role as *models for* (Merz, 2018).

Computer simulations also contribute to epistemic innovation by generating and demonstrating alternative futures and can generate knowledge about functions of material structures that are not yet realised. Computer simulations apply theory, using both abstract theory and mathematical procedures. As epistemic objects, they act as mediators between theory and experimentation; having already defined scientific assumptions embedded into the simulation software on one side and the fundamentally open-ended character of experiments on the other. They are autonomous because of their relative independence from theory and data where they act both as a mediator in between, as well as an instrument for the investigation of both. By encoding different theoretical scenarios into the simulation program and testing their consequences through simulations, the results that are being produced can help identify theoretical assumptions that can be further explored in planned experiments (Merz, 2018).

Computer simulations can be effective tools for both thinking and material interventions, generating both new questions and reliable answers. Particularly complex computer models increase the potential for innovation because of their properties as epistemic objects. Their constant unfolding and multiplex character allow the same simulation model to carry out different functions for different actors in the different contexts where they are used. They can have the function of raising new questions as a research object in one context and at the same time generate answers when being used as an instrument in another (Merz, 2018).

2.3.2 Computer simulation of room acoustics

Computer simulation of room acoustics was first introduced in a scientific context by Schroeder in 1962 (Vorländer, 2013, p. 1203), but it was first since the beginning of 1990's when more powerful processor speeds and memory space were developed that room acoustical computer simulation and auralisation became available on personal computers. Since then, algorithms and processing technologies have developed substantially and commercial room acoustical simulation today include the possibility of auralisation with your built in computer sound card (Vorländer, 2011, p. 72).

Room acoustics computer simulation software combines acoustic theory and modelling software to calculate the trajectory and intensity of sound waves as they are spread around the room. Acousticians position a sound source and listening position in a room model and then track the direction and level of the sound wave as it arrives at the listener. From this information the software can create a 3D representation of how the source would sound like for a listener before the room has been built, known as virtual auralisation. Before the existence of room acoustics computer simulations, consultants would run calculations without being able to observe the results in practice. Room acoustic computer simulation software runs simulations in virtual environments and can calculate thousands of reference points in a time that otherwise would not be possible. The advantages of the room acoustics simulation software has made it an integral part of the industry (Forsyth, 2018).

Some of the basic concepts of room acoustics computer simulation is modelling, simulations and auralisations. *Modelling* is when the acoustic behaviour of a component is predicted by rather limited efforts, simulations and auralisations are when an acoustic object is described in a more complex way (Vorländer, 2020). *Simulations* present acoustical problems in the signal domain, while *auralisations* present it as sound by processing sound signals into an audible result (Vorländer, 2011, 2020). Auralisation makes it easier for people to compare sound signals more precisely through listening (Vorländer, 2020). *Computer simulations* are when sound recordings and input data such as material properties of a room is processed by using geometric models and other methods to solve an acoustic problem and predict a sound result (Vorländer, 2013, p. 1206).

Computer simulations come with some uncertainties and limitations (Vorländer, 2013, p. 1203). Even environmental conditions such as temperature, pressure and humidity affects how sound behaves in a room (Vorländer, 2013, p. 1212). The accuracy and performance of room acoustics computer simulations is validated through modelling existing rooms and then comparing the results with measurements in the actual room. Auralisation can be validated through listening tests where the simulated sound is compared with recordings done in the actual rooms (Vorländer, 2013, p. 1206).

Uncertainties in room acoustical computer software can be due to shortcoming in the algorithms and modelling approaches. A central challenge is to find the appropriate level of detail, whereas too much detail causes unnecessary long computing times and it's necessary to compromise between accuracy and effort. Curved surfaces for instance, are very detailed, making simplifications necessary. When sound hits an obstacle, it is spread in very complex ways. Uncertainties can come from unprecise input data such as models and material properties that come from databases, textbooks, manufacturing variations or product specifications. But also from statistical simplifications and predictions used to overcome too much detail by simplifying the models, or used to overcome uncertainties in input parameters by compensating for these (Vorländer, 2013, p. 1207).

Tests have documented the uncertainties of computer simulations of room acoustics by comparing acoustic simulation software against measurement data. The test showed that simplifications in the underlying theories used in room acoustics simulation software limits the validity of predictions outside limited frequency ranges. Further, auralisations showed plausible but not authentic auralisations, and differences between auralisations of room models in the software and sound measurement in the actual room were clearly audible. The results were accounted to the simplified use of statistics and probability to compensate for complex sound phenomenon, and to shortcoming in models of how sound spreads around obstacles (Brinkmann et al., 2019).

2.4 Summary

This chapter has reviewed relevant literature on software development methodologies and knowledge sharing, as well as the role of computer simulations as epistemic objects. Finally, literature on room acoustics software development was reviewed to understand the complexity and challenges involved in developing such software. We have seen that engineering work consists of practices of knowledge sharing and knowledge development and needs to be observed in specific contexts, such as acoustical engineering. At the same time such practices will necessarily take different forms in specific types of engineering.

In this thesis we contribute to the field by exploring how a small team of mainly acoustical engineers practise the development of complex computer software. By an empirical grounding of object dynamics in specific contexts of expertise, epistemic practices can be advanced. This can bring new insight into how expert knowledge sharing and development happens in work with object construction.

3. Theory

3.1 Characteristics of expert communities

The concept *expert community* is related to the special place of knowledge in contemporary society. The growing importance of knowledge has been recognized as more and more work tasks become specialised and require expert knowledge, and the term knowledge society has been used to describe these current realities (Knorr Cetina, 1999, 2001). With this development knowledge practices originating from the realm of sciences have spilled over into wider society. Still, it is not a new practice, as even renowned engineer achievements centuries ago involved the expertise of both professional experts and locals who were the ones who knew how to make things work in practice. A prominent feature with expert communities today is that they engage in knowledge practices similar to those of scientific institutions in order to develop knowledge for practical appliance (Knorr Cetina & Reichmann, 2015).

Expert communities form around knowledge problems linked to specific areas of expertise, and take part in processes of learning and knowledge creation linked to wider knowledge cultures specific for their expert area. Application of expertise is a characteristic of professional experts, but experts are also producers of knowledge by taking part in knowledge development practices that are part of wider fields of expertise where they produce expert knowledge and integrate knowledge principles in their practical work and application. At the same time professions embark into the realm of science attempting to combine the principles of science and expert work which may create new cultures of knowledge creation. As the sciences, such as engineering, have evolved and become more interdisciplinary and interventionist, knowledge is more distributed and more diverse groups of actors engage in the knowledge development. This means that more knowledge is being developed through the combination of research and practical application in expert communities that are part of expert systems linking the wider areas of expertise (Knorr Cetina, 1999, pp. 1–25; Knorr Cetina & Reichmann, 2015).

When an expert community engages with and tries to find solutions to knowledge challenges or problems, they introduce more complexity and variation to the area of expertise (Nerland & Jensen, 2012, p. 103). The expert community develops specific ways to create and share knowledge both within the group and with the wider community by exploring, developing and mobilising different knowledge objects (Nerland & Hasu, 2021, p. 66) which are central in the work of expert communities (Knorr Cetina, 2001, p. 1). Working on the objects is a driving force for the expert communities to develop their understanding of the problem and the field of expertise itself (Werle & Seidl, 2015, p. 70). The expert communities use specific logics and ways of organising involving tools, artefacts and institutional arrangements together with strategies, visions and procedures to manage and coordinate their work. These logics and ways of organising shape the work of the expert communities and how they become familiar with the knowledge that constitutes their expertise (Nerland & Jensen, 2014, p. 621).

3.2 Expert communities as object-centred

Experts are highly object-centred, and objects dominate much of their work (Knorr Cetina, 2001). In epistemic objects problem areas become visible for expert communities to form around them. They are drawn towards the unsolved problems and questions in the object's material and symbolic expression. The expert communities bring together different tools, ideas, strategies and ways of working and collectively try to solve the problems. When we look at the different epistemic objects that the expert community works on we get insight into how they address the tasks and problems (Nerland & Jensen, 2012, p. 105).

Rather than the experts being centred on the objects, the expert community forms around the epistemic objects in different constellations and there can be different expert communities working on the same area of expertise. The expert communities can be temporal to solve a given time-bound problem, or permanent communities to which the expert feels a strong and emotional tie. The knowledge issues they form around can also exist across different contexts and levels in society. From observing these communities we can get a dynamic understanding of how epistemic objects lead to the formation of different expert communities (Nerland & Jensen, 2012, p. 116).

There is a social dimension of expert communities forming around epistemic objects. The communities make participation possible and contribute to shaping the identity of its members. Identity formation can be similar to the one that takes place in social institutions and the expert communities support the development of stable identities reminiscent of that of a community or family. Thus a result of increasing specialisation in society can be the existence of more object-focused identities and communities (Nerland & Jensen, 2012, p. 113).

The objects, practices, strategies and visions of expert communities are interconnected through wider machineries of knowledge construction. They explain how different responsibilities are distributed through extended networks where efforts of practitioners, researchers and agencies interact in the development and circulation of expert knowledge. By looking at the wider machineries of knowledge construction we can explore connections and relations between epistemic practices across sites and contexts. This can reveal the dynamics of how knowledge practices and culture are interrelated in society today. While some expert communities are nationally bound, others take part in global knowledge networks reaching out beyond national borders (Nerland & Jensen, 2012).

These wider machineries of knowledge construction operate on different levels, such as the macro, meso and micro-level. The *macro* level encompasses the specific actors and organisations that have a role of verifying expert knowledge and take on specific tasks in larger knowledge contexts. These actors bring together knowledge and create standards for the knowledge practices of a domain. They also contribute to networking across different settings in an expert field. The *meso* level is where experts show agency when applying generalised expert knowledge to specific cases. At the *micro* level epistemic objects assist the process of knowing and have a transformative and stabilising role in expert practices. Epistemic objects can connect expert communities with the wider field of expertise and make it possible to observe how these practices take place in complex societies where several network structures exist in parallel and knowledge is constructed on different sites (Nerland & Jensen, 2012, p. 106).

The interactions between actors in the wider machinery of knowledge construction are driven by knowledge dynamics coming from working on the epistemic object. When experts pursue knowledge questions elicited from working with epistemic objects, they get involved in the extended knowledge settings of the expert field. The interactions can be facilitated by intermediaries who check the quality and relevance of the knowledge that is being brought to the epistemic object. There are different logics of knowledge production involved in this that are governed by the expert communities who possess complex knowledge about the field of expertise. The experts work on the object from different perspectives in these wider machineries. You have professional institutions educating new practitioners as well as gathering and developing knowledge on the field. You have conferences and international networks. Different actors and organisations help keep the knowledge object alive by creating infrastructure enabling engagement that is driven by the knowledge dynamics (Nerland & Jensen, 2012, pp. 112–116).

3.3 Knowledge sharing through object construction

Expert communities develop epistemic objects in specific ways though generating and sharing knowledge (Nerland & Hasu, 2021, p. 66). Expert communities' work on epistemic objects can lead to knowledge acquisition, knowledge integration and knowledge sharing. When an expert community gets emotionally invested and shows intimate attachment towards a shared epistemic object it triggers emotional and cognitive trust. Epistemic objects motivate experts to mobilise knowledge flows and increase the continuity in knowledge generation. The motivation comes from an urge to know, and the experts are kept together in joint knowledge work around the epistemic object through shared interest, common goals and a need to know what the others know. By establishing relationships with their epistemic objects and engaging in knowledge processes expert communities can identify with their expert field, leading to more engagement with the field (Liu, 2019).

The epistemic object gives the members of the expert community an overview of each members' expertise which helps them to identify what knowledge they need in a precise way. This mechanism of the epistemic objects to elicit the knowledge of the expert community members makes it possible for the members to identify what knowledge exists in the community. The epistemic objects trigger the development of joint transactive memory in the expert community, understood as the community members knowing who knows what. This increases collective sensemaking and facilitates a merging of the fragmented pieces of information into quality knowledge output. In this way epistemic objects coordinate the heterogeneity of knowledge existing in the expert community and incite the generation of new knowledge (Liu, 2019).

Epistemic objects have a productive and enabling role for knowledge sharing in expert communities. It triggers trust and elicits knowledge by establishing temporal expert communities around knowledge development processes. When experts working on the epistemic object struggle with its materiality and the conflicting strands of knowledge they introduce, they gain motivation, get energy and give direction to the knowledge generation processes. The focus shifts from experts controlling objects as instruments and tools to how the epistemic object and formal management strategies coordinate knowledge that is fragmented and heterogeneous to generate new knowledge. Summarised, the epistemic objects trigger trust and elicit knowledge which lead to knowledge sharing, knowledge acquisition and knowledge integration by coordinating fragmented and heterogeneous knowledge in expert communities focused on generating new knowledge (Liu, 2019).

Computer simulations as an epistemic object can have a special function in contributing to knowledge sharing and can be used to generate both theoretic, implicit and practical knowledge. For the expert communities working on computer simulations, knowledge elicited can be shared and may lead to the generation of possible future artefacts, conceivement of alternative theories or generation of knowledge for specific theories (Merz, 2018).

3.4 Working with epistemic objects

When a community of experts work with epistemic objects, the object is in the centre of knowledge development. This form of work is different from working with technical objects as the epistemic objects have distinct properties and consequences for how work takes place in expert communities (Knorr Cetina, 1997, 2001). Epistemic objects are created when expertise is mobilised to deal with emergent and complex societal problems. They are built on knowledge created by science, but are at the same time oriented towards being used (Nerland & Jensen, 2012, p. 105).

The epistemic object is the point of focus in a process of exploration where experts try to solve the complex knowledge problems it raises. As the experts explore the object, questions are being generated and the object opens itself and its complexity increases as the knowledge of the object is increasing (Knorr Cetina, 2001). They are complex combinations of symbolic and material resources that make up the knowledge surrounding a problem (Nerland & Jensen, 2012). Epistemic objects can be outputs of an experimental process in science which changes as a result of the process, or they can communicate meaning as an input to an epistemic work in a process of exploration (Ewenstein & Whyte, 2009, p. 11).

The experts engage in a process of trying to define and make the object into something concrete and material, but it is not possible to exactly define an epistemic object. By exploring the epistemic object, its lacks and shortcomings become visible for the experts who become painfully aware of them. The epistemic object can be estimated and projected but it is never completed. It is in a process where it is continuously changing and unfolding and there is always something that is incomplete or not understood about it. The object keeps unfolding infinitely and in the process of constantly changing it acquires new properties and changes its existing ones. Because it is incomplete, the epistemic object does not completely resemble itself, and it can't give the full picture of the ideal object it is meant to represent. So when trying to define it, it is not defined so much from how it is represented today, but how it will become at some point (Knorr Cetina, 1997, 2001).

The expert is bound to the object through a creative and constructive practice. The reason for this lies in how the epistemic object unfolds and reveals itself, making the expert want to continue to develop the object. When the lacks becomes visible in the object the object gives suggestions on how it can be unfolded further which gives the epistemic object a signifying force. Signifying in the sense that it leads the expert to explore the object in a new way and engage in meaning producing (Knorr Cetina, 1997). Because of their inner complexity they open up a range of opportunities when being explored. The objects consist of a variety of artefacts and resources that are interconnected and hold multiple opportunities for being explored and used (Nerland & Jensen, 2012).

Through the process of unfolding the epistemic objects become divided into many partial objects or instantiations of the objects. The partial objects represent the object, but as the epistemic object continuously changes to become what it lacks, new temporary partial objects are being made. Each of the instantiations are true epistemic objects as the epistemic object can exist as a variety of forms and the many partial objects are always in relation to the epistemic object as a whole. They have the same quality of being able to unfold as any other partial object (Knorr Cetina, 2001).

The signifying force of the existing partial objects drives the creation of new epistemic objects by giving directions and demonstrating its needs for further development. The signifying force of partial epistemic objects directs future exploration and leads to the generation of new meaning and practice. It signals constructive ways of extending practice by opening up strings of possible paths forward that experts can follow (Knorr Cetina, 1997, 2001).

It is in the form of a partial object where the investigation of epistemic objects takes place and where subjects can interact with the object. The epistemic object itself remains reachable only through partial objects. The representation of the epistemic object is available for the subject to manipulate and explore its properties. They represent something larger than the object itself such as knowledge that is not yet known and as objects of exploration they generate interest and motivation (Werle & Seidl, 2015, p. 70). The epistemic object changes as its representation is changed, and the material instantiations are thus a crucial dimension where practitioners interact to develop knowledge. By being both, epistemic objects bridge the dimensions between the abstract epistemic object and its concrete material instantiation (Ewenstein & Whyte, 2009, p. 11). Partial objects help understand the function of the object when working with epistemic objects. It functions both as a mediator and a material instantiation which can be used to work out ambiguities and generate meaning, but also creates some limits to the dynamics by coordinating the experts input and allowing for some parts to be stabilised while others are being developed. It can also be used to exert control by defining what practices are internal and which take place publicly, who control the use of the object and who are being held accountable (Tronsmo & Nerland, 2018, p. 35).

Epistemic objects will need to be materially defined in order for professionals to explore and develop them further. By materialising the epistemic object, it is stopped from continuously changing. This can be done by ignoring the lacks and discrepancies of the object. It can then gain a material identity making it possible to use it for a particular purpose. Computer software can be used as an example of how experts can stabilise objects. Computer software is constantly changing and unfolding as experts write, run and update code according to their changing interests. Simultaneously they serve a user community and issue different versions, updates and editions of the code for them. The different modifications made to the code are compiled into versions and updates that are familiar to the users and the developers take special efforts to meet the users' needs (Knorr Cetina, 2001).

When an object is materialised it takes a more fixed identity as closed, determined and defined. It is no longer a problematic object, but takes the form as a tool that can be applied in practices, taking the state as a technical object rather than epistemic and question generating (Ewenstein & Whyte, 2009, p. 12). Epistemic things can be transformed into technical objects and act as an infrastructure for further innovation and knowledge generation. Computer simulations spread innovation processes over time, actors and contexts who engage with the simulations, making its stabilisation partial and temporary (Merz, 2018, p. 336). Epistemic objects are complex artefacts and act as both a tool and an object of investigation in a concrete epistemic practice. They can be used both for exploration as well as an instrument, and switch between the two in processes of exploration and confirmation. Epistemic objects influence how expert communities develop knowledge, giving way to epistemic practices (Nerland & Jensen, 2012).

3.5 Epistemic practices

Expert communities engage in epistemic practices when working with epistemic objects. Epistemic practices take place when an expert is faced with a new problem which makes them have to relate to an object in a different way. While many daily uses of objects are automatised and routine, epistemic practices start working when the relation between a subject and object becomes disconnected and non-routine (Knorr Cetina, 1997, 2001).

Epistemic practice comes from the separation between subject and object and is a form of dynamic relations that can be understood as disruptive, creative and constructive. It is in contrast to the routine, rule based and iterative practices that come from using objects in a technical and instrumental way. The dynamic relationship drives the transformation of both the epistemic practice and its epistemic object (Knorr Cetina, 2001). The relationship may also be multi-actor, and not just between one person and an object (Ewenstein & Whyte, 2009, p. 12) and epistemic objects become further developed and passed around as experts in different contexts engage with them though exploring their complexity and materialising their potential uses in their own activities (Nerland & Jensen, 2012).

An epistemic practice focusing on epistemic objects enables experts to explore complex problems, access knowledge resources, and identify open opportunities and temporal solutions. The flexibility of epistemic objects to take on different roles in a system of experimentation supports their continued existence. They have the ability to structure practice and give directions on how they can be used and further explored. They demonstrate what experts do with knowledge and how they engage to develop knowledge and practice. Here work shifts between exploring and confirming practices such as exploration and testing (exploring practices) and validation and documentation (confirming practices). Exploring practices contribute to the unfolding of the epistemic object. They open up new possibilities for elaboration and project new partial objects and possibilities. Confirming practices on the other hand, helps stabilise and fix the object, making temporary materialisations possible. Together they make it possible for experts to handle complex challenges (Nerland & Jensen, 2012).

3.5.1 Epistemic practices in engineering

In engineering Cunningham & Kelly (2017) identifies 16 epistemic practices that cover important aspects for engineering. They divide them into a framework of four partially overlapping and interconnecting categories; *engineering in social contexts*, *uses of data and evidence to make decisions*, *tools and strategies for problem solving*, and *finding solutions through creativity and innovation* (see Table 1).

The first category is *engineering in social contexts* and refers to how engineering problems and knowledge originate from social needs and are solved through social processes, including evaluation by clients. The practices can be external or internal. External is when the problems and solutions take place in social settings external to the internal processes of the team, and where the problems are situated in varied contexts and dependent on local conditions. Internal is the ways engineering takes place socially within the team. Internal practices deal with the inner workings of engineering teams. Her most work happens as collaboration among team members. Engineers work based on knowledge, practices and norms of the community and generate social knowledge through social interaction based on the community norms and existing knowledge (Cunningham & Kelly, 2017).

Categories	Practices
Engineering in social contexts	*Parameters Set by Clients and Conditions.* 1 Working with problems in context 2 Making trade-offs between criteria and constraints 3 Assessing the implications of solutions in the real world *Inner Workings of Engineering Teams* 4 The need to communicate effectively 5 Working well with teams 6 See themselves as engineers 7 Persist in the face of failure
Uses of data and evidence to make decisions	8 Making evidence-based decisions 9 Constructing models and prototypes
Tools and strategies for problem solving	10 Applying mathematical and scientific knowledge to problem-solving 11 Envisioning multiple solutions, 12 Considering materials and their properties, 13 Using systems thinking 14 Building and learning from prototypes
Finding solutions through creativity and innovation	15 Developing processes to solve problems; 16 Innovating processes, systems, and designs;

Table 1: Epistemic practices in engineering

The second category is *uses of data and evidence to make decisions*. Engineering is highly empirical and interaction with the world, and data and evidence is fundamental for effective engineering. Evidence is central to how engineers construct knowledge to solve problems by understanding users' needs, assessing problems or challenges, testing parameters, building effective prototypes, testing models and designs in context and presenting solutions to clients. The third category is *tools and strategies for problem solving* and refer to the evolving tools and strategies used in problem solving, providing analysis and designing solutions. These are used in epistemic practices that lead to knowledge being embodied in solutions. Engineers define problem solving as central to engineering. The fourth category is *finding solutions through creativity and innovation*. The reasoning processes in engineering are highly context dependent, making creativity important for innovation and finding solutions. Many of the other practices connect with this.

Cunningham & Kelly (2017) find that epistemic practices have four characteristics. They are *interactional* and constructed among people through joint activity. They are *contextual* and situated in social practices and cultural norms for presenting, representing and assessing knowledge claims. They are *intertextual*, and semantic representations are referenced, reviewed, appropriated, interpreted and reinterpreted through social practice in coherent and interactionally recognised ways of communication. They are *consequential* and how knowledge is created, represented, evaluated and legitimised have consequences for what and whose knowledge counts when the practice is recognised in a related epistemic culture.

3.6 Summary

This chapter presents a theoretic framework based on conceptualisations of expert communities, epistemic objects and epistemic practices in the works of Knorr Cetina and related theorisations within professional and engineering work. First, we explained what expert communities are, how they are object centred and share knowledge through object construction, further we explored what working with epistemic objects entails and how they happen through expert communities engaging in epistemic practices. We will here summarise the concepts and present how they will be used in the analysis.

Expert communities form around knowledge problems linked to specific areas of expertise and engage in knowledge practices similar to those of scientific institutions to develop knowledge for practical appliance. Combining scientific and expert work they create new cultures of knowledge creation using specific logics and ways of organising that shape the work and expertise of the community (Knorr Cetina & Reichmann, 2015; Nerland & Jensen, 2014).

Working on objects is a driving force for expert communities to develop their understanding of the problem and the field of expertise itself. Expert community forms around the epistemic objects and observing these communities help us understand how this happens. The knowledge dynamics from working on the epistemic object drives the interactions between the actors in the wider machineries of knowledge construction (Nerland & Jensen, 2012).

Epistemic object work is in the centre of the knowledge work of expert communities. Epistemic objects are the focus of complex problem solving and are based on scientific knowledge, but oriented towards use. Their distinct properties have implications for how work takes place (see Table 2).

Characteristics of epistemic objects
• Generate questions and increase in complexity as they are being worked on.
• Are always in a process of being defined, where new lacks become visible as they are being worked on.
• Are incomplete, constantly unfolding and changing its properties, in the process of becoming what it is not yet.
• Engages the experts in constructive practice where it leads to new opportunities and ways of exploration and meaning producing.
• Becomes divided into many partial objects as they unfold, each having the qualities of the epistemic object and standing in relation to it as a whole.
• Has a signifying force of giving directions and showing needs, driving the creation of new partial epistemic objects and practices.
• As partial objects function as a mediator and material instantiation that experts can interact with to work out ambiguities and generate meaning.
• Needs to be materially defined and stopped from changing to be used for a particular purpose, it can then be used as a tool applied in practices.
• Switch between being used as an epistemic object and technical tool through the processes of exploration and confirmation.
Sources: (Knorr Cetina, 2001; Nerland & Jensen, 2012; Ewenstein & Whyte, 2009; Werle & Seidl, 2015; Tronsmo & Nerland, 2018).

Table 2: Characteristics of epistemic objects

Epistemic practices are the specific practices and dynamic relationship experts engage in when developing knowledge and practice, or what experts do when working with epistemic objects. Epistemic practices enable experts to explore complex problems, access knowledge resources and identify open opportunities and temporal solutions. The epistemic object is flexible to take on different roles as an object of exploration which structure practice and give direction for future exploration (Knorr Cetina, 2001; Nerland & Jensen, 2012).

Knowledge sharing and generation is central to the specific ways expert communities develop epistemic objects. Driven by an urge to know, epistemic objects motivate experts to take part in joint knowledge work, mobilise knowledge flows and increase the frequency of knowledge generation (Liu, 2019). Computer simulations contribute to knowledge sharing through their own dynamics of innovation where they raise new questions and answer existing ones, acting as a mediator between theory and experimentation (Merz, 2018).

The concepts presented in this chapter compromise an analytical framework that first will be used to inform categories and structure the presentation of our data in the analysis, demonstrating how expert communities share and develop knowledge by working with epistemic objects, and second to discuss how our findings relate to and inform the theoretical concepts in the discussion.

4. Methodology

This chapter presents our empirical work and approaches to the data collection by justifying our assessments and presenting our reflections during the research process, with the purpose of presenting our work as transparent and credible.

The aim of this master thesis is to explore how knowledge sharing and knowledge development takes place in highly specialised knowledge work. To increase our understanding of this we based our empirical work on a team of experts that collaborated on developing software for room acoustics and studied how they as an expert community shared and developed knowledge, and what challenges the team faced while working on their joint epistemic objects.

The subchapters below present our methodological decisions and assessments and are based on relevant methodological literature to provide a thorough foundation to answer our research problem.

4.1 Research Design

The choice of research strategy, design and method must be tailored to the research question being investigated (Bryman, 2016, p. 36). Given the aim of our study we considered that we would benefit from a qualitative approach because qualitative research is a strategy that emphasises words rather than quantification in the collection and analysis of data (Bryman, 2016, p. 374) and as we were interested in understanding *how* knowledge development and knowledge sharing took place in the particular expert community that we studied, and in *what ways* epistemic objects and practices contributed to the development of knowledge, we considered that this approach would provide us with relevant and useful data through semi-structured interviews and observations.

Case study design

Our study is based on a company composed of seven experts and one secretary collaborating on developing a comprehensive software solution for room acoustics. The software has existed for almost 40 years, and the years invested in the development has provided a reliable software with a wide range of applications developed by the team of experts. Given their history and background we considered the company qualified as a case that could provide an environment where we could investigate this particular expert community and how the development and sharing of knowledge were facilitated by working with joint epistemic objects.

Choice of methods for data collection

The research design helps present a structure of the topics to be focused on, relevant participants for interview and observations, as well as an idea of where and how the interviews and observations are to be carried out (Thagaard, 2018). Our aim was to get a deeper understanding of the team's experiences and thoughts regarding their knowledge sharing activities when working with their joint epistemic objects. To collect such data, we decided to start our data collection process with an initial semi-structured interview with all eight employees of the company to familiarise ourselves with their educational background and areas of expertise individually, and how they experienced their own practices and interactions as a team.

The research process for qualitative research is emergent, meaning that the initial plan for the research cannot be followed to the exact detail (Creswell, 2013, p. 47). As we had finished the initial rounds of interviews, we discovered several objects that we considered interesting and wanted to follow up on. We therefore developed a follow-up interview guide and planned to interview four members of the team who were directly involved in the development of the objects of interest.

The key idea behind qualitative research is to learn about the problem or issue from participants and engage in the best practice to obtain information (Creswell, 2013, p. 45). We considered it necessary to develop a research design that allowed us to gain a thorough understanding of the team's knowledge sharing activities and how those activities could be understood through working with epistemic objects. To gain a broad understanding of the teams' work we conducted both interviews to get the informants' own descriptions, and observations to get a broader understanding of how their work unfolded in real life.

We made six observations of the team's weekly meetings over a period of six weeks. A potential challenge in observational studies is to be able to choose a relevant time and place where the researchers can collect observational data relevant to the research problem (Tjora, 2017, p. 55). Because of travel restrictions in conjunction with the Corona pandemic we decided to conduct the observations of the weekly meetings digitally and observe via zoom, like we had done with the initial interviews. Yet, qualitative researchers often collect data in the field where the participants experience the issue or problem under study (Creswell, 2013, p. 45), so with this in mind, as soon as the travel restrictions were repealed, we had a one week visit to the company's office to conduct the follow-up interviews and be able to observe the team while being physically present. We felt strongly that this was important to prioritise given that one of our main interests in this study was to get an understanding of how the team interacted with each other and shared and developed knowledge. We also saw it as an opportunity to be able to discover potential non-communicated knowledge sharing activities that the team perhaps were not observant of themselves or knew how to explain.

Research methods such as participant observation and semi-structured interviewing are used so that the researchers can keep an open mind about the shape of what they need to know about, so that the concepts and theories can emerge out of the data (Bryman, 2016, p. 10). We organised our empirical work as shown in Table 3, by conducting both initial and follow up interviews, as well as making observations and collecting documents.

No.	Interviews	Type of communication	Time
8	Initial interviews	Digital	45-60 minutes
4	Follow-up interviews	Physical	30-45 minutes
1	Group interview	Physical	60 minutes
	Observations		
6	Observations of weekly meetings	Hybrid	1-2 hours
	Informal observations at the company's office	Physical	3 days
	Documentation		
	Pictures, documents, and screenshots		

Table 3: Data collection

4.2 Selection of case and recruitment of participants

Qualitative studies can be characterised by a limited number of people and because of the relatively small sample it's important that the selection is well adjusted to the research problem, so that the analysis can provide an understanding of the phenomena we are interested in (Thagaard, 2018, p. 54). We recruited our participants based on their suitability as a team, as we see them as a true expert community collaborating on developing a highly specialised software for room acoustics, and from our point of view this particular selection could provide us with relevant insight and empirical data that could contribute positively in answering our research questions, as well as our broader research problem.

Case description

Our case company produced a simulation software for room acoustics that was sold and distributed globally and was initially started as a university project in the 1980's. The software was developed for acousticians and building consultants and its functions can roughly be divided into two main areas consisting of room acoustic simulations and sound measurements.

The software is based on scientific theories and international standards of room acoustics, and has been constantly developing for 40 years with new improved versions, new functionalities following the development in scientific theories and the needs of the users.

The team mainly consisted of employees with a masters or PhD in acoustics, a specialisation within engineering studies. Within the team there was also a person with a high level of competence within the fields of astronomy and astrophysics and mathematical modelling. In addition, there was a secretary responsible for sales and marketing.

The team was composed of a manager, the founder, a secretary, four employees and a student assistant. Their main tasks were to work on areas such as research and development, providing technical support and making video tutorials and instructional videos for the users of the software. The employees produced scientific papers and participated in international acoustical conferences. The company also had a board of four, representing consulting companies who are also users of the software. Due to the tight connection with the university, the company provided free software access for students and staff for educational purposes in exchange for access to the acoustics facilities of the university.

4.3 Approaches to data collection

To gain a broad understanding of how the company was structured and how they organised their work, we decided to interview all the employees, and collected a thorough empirical database composed of several semi-structured interviews and observations. We prepared various versions of interview- and observation guides and worked our way systematically through all of them to ensure that we covered all the relevant parts of the team's activities.

We ended up with two thoroughly planned interview guides, one for the initial round of interviews (see Appendix D), and one for the follow up interviews (see Appendix E), in addition to one observation form (see Appendix G) that we used both for digital and physical observations.

a) Interview guide

As we were interested in gaining rich descriptions from the team members we decided on a semi-structured interview, which is usually referred to as an approach that allows us to prepare more general questions and allows us as interviewers some latitude to ask further questions in response to what we see as significant replies (Bryman, 2016, p. 201).

As the team used English as their preferred work language, we decided to do so as well, and prepared our interview- and observation guides accordingly. Since we collected our data material in English, we further decided to write our thesis in English to preserve the essence of the data material and provide accurate quotes and descriptions from the informants. We chose to structure the interviews thematically to ensure that we covered all the relevant areas of their work. Mainly we wanted to capture the informants' own descriptions on how they experienced their knowledge sharing activities and how the professional interaction among them as experts contributed to the development of their shared epistemic object.

We structured our interviews based on four topics with associated sub-questions, using the same topics both for the initial and the group interviews. The topics were: *introduction/background, team organisation, knowledge sharing*, and *epistemic objects*. In the follow up interviews, we adjusted the topics to focus on the development of the epistemic objects we wanted to explore further. The topics for follow-up interviews were: *origin (of the object), team collaboration, knowledge sharing and epistemic practices*.

We decided to start the initial and group interviews with open-ended questions to get broad insight into the informants' educational background and a better understanding of their areas of competence and expertise. We started the interviews with topics that could be perceived as "safe" and hopefully seem easy to talk about, to facilitate a comfortable flow and safe environment throughout the interview. We considered it appropriate to follow up with a topic concerning them as a team to gain an understanding of how their educational backgrounds and areas of expertise worked together, and how they organised and coordinated their knowledge as a team. Considering our research problem, we decided to include the topic of knowledge sharing in our interview guides with the purpose of collecting rich descriptions of knowledge sharing activities from the team. We also focused on providing questions that allowed the informants to elaborate on potential challenges they had faced when developing the software. The fourth and final topic referred to epistemic objects. We considered it appropriate to introduce this topic at the end of the interview as we thought it would be useful to have gained the insight and knowledge from the previous themes as background information to better understand the description related to the work with epistemic objects.

b) Conducting interviews

As part of the preparations for the interviews, we conducted a test interview to get as relevant results as possible from the actual interviews. The test interview was conducted with an informant who had a background from designing and developing epistemic objects to test whether the questions we had prepared were understandable and possible to answer from a technical design perspective. The purpose of the test interview was to experience how it would be to conduct the interview as we had planned it in our interview guide. Our main focus during the test interview was the time frame and general flow throughout the interview and attention to structure and the formulations of our questions. The test-interview revealed that we would benefit from changing some questions to become more precise. As a result of the test-interview, we went through the interview guide and identified some changes that needed to be made and did the necessary adjustments to be better prepared for conducting the initial interviews.

Initial interviews

Our case company is a foreign company with offices located abroad, and corona restrictions were still in place at the time of data collection. This made it necessary to conduct the initial interviews digitally. We created a digital timetable where the employees could book a suitable time for their interviews, and we used Zoom to conduct the interviews. Each interview was estimated to last from 45 minutes to an hour.

We thoroughly planned how to conduct the interviews and had two main areas of responsibility during the interviews. One was to welcome the informant and introduce us and our research project. This role was in charge of paying attention to interesting new topics that we could pursue, whereas the other role was in charge of leading the interview following the interview guide. During our first interview we discovered that some of the question formulations in our guide would interfere with the flow of the interview. After the first interview we sat down and tried to identify and reformulate the questions that had caused the problems. We experienced during the second interview that we had managed to identify the formulations that caused the problems and we did not experience the same misunderstanding that we did in the first interview. Based on this we updated our interview guide for the upcoming interviews. These changes can be seen in the initial interview guide (see Appendix D). After each interview, we sat down and shared our immediate thoughts and wrote a short report with the highlights and potential paths for us to follow. As we had conducted several interviews, we discovered that our knowledge on our informants' activities had increased and we experienced a saturation in the answers, where for each interview less information was new to us. This led to an overall better understanding and made it possible for us to ask more relevant follow-up questions. Considering our research problem this broader insight also led us to finding interesting and relevant paths that we decided to pursue with follow-up interviews.

Follow-up interviews

From the initial interviews we hoped to identify specific examples that could illustrate the processes of how knowledge is shared and developed in this expert community. We identified two particularly interesting objects that we saw as relevant contributions in answering our research problem. These were two different types of objects, one being a software feature and the other a hardware stand-alone object. Both shed light on interesting aspects of knowledge sharing through their function as partial epistemic objects.

We contacted the informants who were responsible for the development of each object and planned for a follow-up interview. The purpose was to gain further insight on the development process, and get detailed information on how the developers documented their work, as well as rich descriptions on their knowledge sharing processes when working on their respective objects. We decided to conduct four follow-up interviews, and created a new updated guide (see Appendix E) specifically tailored to the objects we wanted to examine. Based on the main topics from the initial guide we developed new questions that would facilitate detailed descriptions of the object, as well as the informants' own experiences and thoughts from having developed these objects.

Group interviews

Our team of experts were a small, specialised team with a high level of knowledge and competence within their fields of expertise, and they depended on a high level of cooperation and close communication to manage some of their more complex tasks. In order for us to gain valuable insight on the team's everyday work we decided to conduct a group interview where everyone was physically gathered to learn more about them as a team, and hear their descriptions of the teams' cooperation processes when solving complex challenges together. We used the same template (see Appendix F) as we had used in the previous interviews with four main themes - background, team, knowledge sharing and epistemic objects. We chose to use the same themes as a basis throughout all our interviews to provide a structure that we could benefit from when we were to systematically analyse our data and compare answers from different interviews within the same topics. Similar to the follow-up interviews, the group interview was conducted as a part of our visit to the company's offices. We sat together in their meeting room and completed the interview within one hour. Every team member except one (who worked remotely from abroad and could not attend physically) participated in the interview, which provided us with interesting and relevant data for further analysis.

c) Observation

We decided to observe the teams' weekly meeting over a period of six weeks so that we could follow the development of the team's many projects over time. We focused on observing how they as an expert community shared knowledge that contributed to the development of their joint epistemic objects. We considered it useful for us to observe several meetings so that we could follow the team over time and get a better understanding of how their projects developed and observe in practice how the team proceeded to solve complex challenges that occurred during the development of the objects. The observation gave us a more holistic understanding of how the team worked by letting us observe them in a natural working setting.

We prepared for the observations in the same way as we did with the interview guides, and developed a template (see Appendix G) by discussing relevant themes that we found interesting to investigate further. We came up with four themes that we used to categorise our observations with suitable subthemes to focus on during observations. The main themes were: *organisation of work, relations and dynamics, tools* and *objects, and knowledge sharing.*

We used these themes as a basis and wrote individual notes that we compared after each observation trying to identify key findings. We conducted a total of six observations, whereas five were digital and one where we were physically present at the office.

Digital weekly meetings

We observed the majority of the meetings digitally using Zoom. We received a link in advance that provided us with easy access to the meetings. We started our observation a week later than planned due to a technical issue, but with that issue sorted out we were ready to start our observations the week after and the team had more time to get accustomed to the idea of our presence. The team was already used to other team members attending the meetings digitally, so to our knowledge, our digital presents did not interfere with the way they usually held their meetings. As the video screen in the meeting room only showed the last speaker, our visibility was low in the meeting room where most attended from.

During our first observations we did a short presentation introducing ourselves and our project for us to get to know the team and for them to get to know us. We also wanted to provide the team with the opportunity to ask questions about the project. After we had finished our presentation and answered questions from the team they proceeded with their meeting as usual, and we observed as planned.

Physical weekly meeting

During our visit to the company’s office we also conducted a physical observation. Even though we were present in the room we kept some distance and tried not to interfere with the meeting. It was the third day of our visit to the company and we had gotten better acquainted with most of them through joint lunches and follow-up interviews. We experienced that they were getting used to us being present as they took less and less notice of us, but we were aware that our presence also could affect their behaviour during the meeting. Despite this we found that the team were fully focused on their meeting schedule and the meeting followed the same structure and dynamics as during digital observations. For our part we conducted the observations in the same way as when we observed digitally and took notes that we compared and discussed after the meeting.

d) other data sources

Other data sources that we collected were pictures that we took during our stay, which the team were informed about and agreed to. We also received models, sketches and documentation that provided us with more information on the objects that we were particularly interested in, as well as access to the managers internal meeting schedule so that we could follow the progress.

4.4 Analytical Approach

The analysis process involves converting the collected empirical data from oral to written data material (Thagaard, 2018). We converted our empirical data from oral to written by transcribing the interviews, we also coded our transcribed data with the purpose of achieving a comprehensive understanding of the themes we have studied, and facilitating a systematic analysis process.

Audio recordings

Audio recordings provide the most comprehensive information about the dialog between the researcher and the informant (Thagaard, 2018, p. 111). We made informed audio recordings of all of our interviews, both the ones we conducted physically and digitally. During our digital interviews and observations, we recorded the audio from the video conferences conducted through Zoom, and stored the audio files in accordance with ethical guidelines and general data protection regulations (GDPR). The advantage of recording is that everything being said during the interviews is safely preserved which provides us with the opportunity to fully concentrate on following the interview guide and pay attention to the informant's answers and ask relevant follow-up questions. Audio recordings are only used when the informants have given their permission (Thagaard, 2018, p. 112). We collected consent (see Appendix C) through an inquiry that we sent out in advance where we gave comprehensive information regarding the data collection process and their rights as participants in our project (see Appendix B). In addition to this, we also informed the participants that we made audio recordings and their right to withdraw their consent at any time prior to each interview.

Transcribing

Transcription is an important part of the analysis process as it is the most effective way of converting our data from oral to text material (Thagaard, 2018, p. 112). Our initial plan, having recorded every interview, was to listen to the file and write it into a transcribing program. This turned out to be a challenge, and we tried several methods for transcription. We started to manually transcribe some interviews directly from the audio file, but discovered that this was both time-consuming and presented challenges as we at this early stage did not have a comprehensive understanding of our informants' professional language. This caused challenges during the transcribing process as some descriptions were hard to comprehend. As a result of this we started using NVivo (*NVivo*, n.d.), a tool that complements the work of researchers using qualitative methods in which we also did all our coding. Even though we spent some time learning how to use NVivo, it turned out to be a time-saver, and allowed us to save valuable time from analysing the content manually and gain a comprehensive understanding of our collected material.

We divided the transcription of the interviews equally between us and decided that each should transcribe the interviews where we did not have the leading role ourselves. Although we were both present during all interviews, we made this decision because we wanted both to be involved in all interviews, either by leading or transcribing them. All identifying information such as names of employees, the company name or name of the software were removed from the transcripts, and the names of the team members were replaced with pseudonyms. We decided to use pseudonyms to preserve the personal approach the team had to their projects while keeping them anonymous.

The initial interviews lasted about an hour and the follow-up interviews varied between 30 minutes to an hour. We wanted to preserve as much of the content and descriptions as possible and we therefore chose to transcribe all the interviews in their entirety to ensure that we did not miss out on important information, which also provided us with a comprehensive basis for the analysis.

Thematic analysis

To analyse the collected data material, we have chosen to use thematic analysis, which is a method for identifying, analysing and reporting patterns (themes) within data (Braun & Clarke, 2006, p. 6), see Table 4.

Phase 1:	Familiarising ourselves with our data	Transcribing the data material, get an overview of potential topics
Phase 2:	Generating initial codes	Create codes based on interesting aspects of the data material and gather relevant data within the codes
Phase 3:	Searching for themes	Create potential themes by gathering corresponding codes
Phase 4:	Reviewing themes	Reviewing and refining themes, check whether the themes work in relation to both levels. L1: the coded extracts L2: the entire data set. Provide thematic overview of the analysis
Phase 5:	Defining and naming themes	Refine the essence of the themes that will be presented in the analysis, and analyse the data using those detailed themes
Phase 6:	Producing the report	The task of the write- up of the report is to tell the complicated story of the data in a way that convinces the reader of the validity of the analysis.

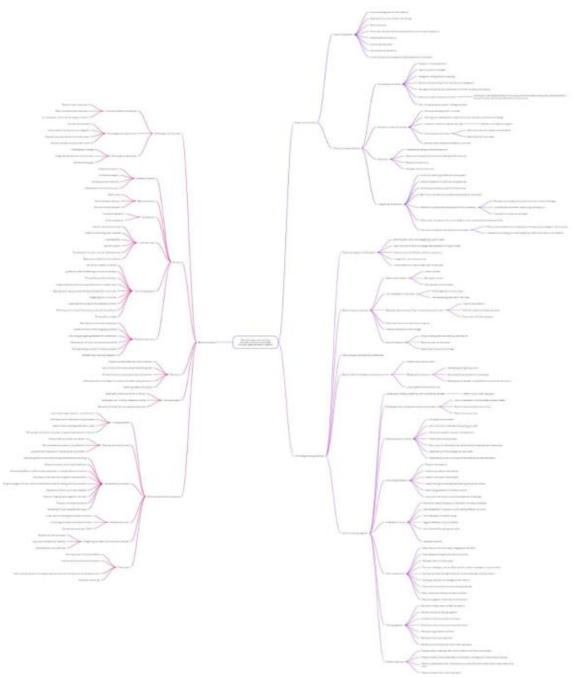
Table 4: Thematic analysis

Familiarising ourselves with our data

The first phase of a thematic analysis involves familiarising ourselves with our collected data material. As described by (Braun & Clarke, 2006, p. 16), it was vital that we immersed ourselves in the data to the extent that we were familiar with the depth and breadth of the content. We started this process after the completion of each interview where we wrote down our immediate reflections and discussed them with each other. After we had completed all our initial interviews, we started the process of transcribing and familiarising ourselves with the material through text. According to Braun & Clarke (2006) it is ideal to read through the entire data set at least once before coding it, as our ideas, identification of possible patterns will be shaped as we read through it. We both read all transcriptions so that we became familiar with the content of each interview and could detect relevant patterns for coding.

Generating initial codes

Once we had gotten to know our data material through transcribing, we proceeded with the second phase, which is presented by Braun & Clarke (2006) as the production of initial codes from the data based on features that appeared interesting to us. Building on the work we had done in the previous phase we started generating initial codes individually, using mind maps (see Picture 1). Further we compared our work and agreed on codes that seemed relevant and continued developing more codes together which finally provided us with an overview of our material by organising our data into what Braun & Clarke (2006) describe as meaningful groups.

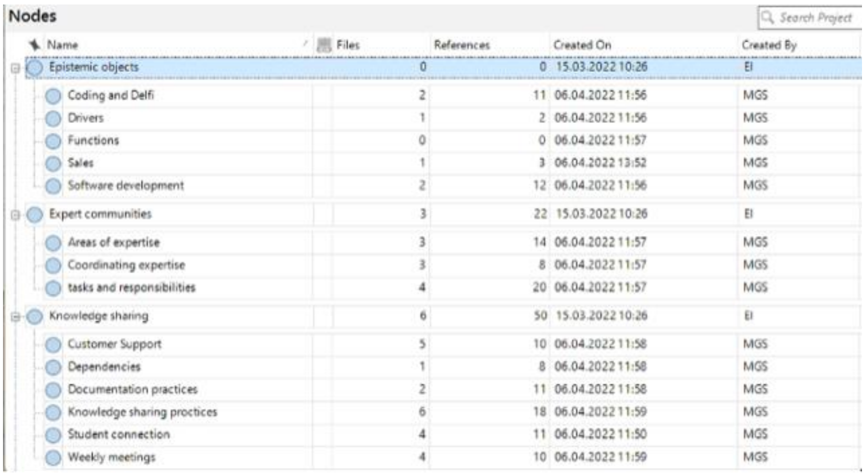


Picture 1: Initial codes

Searching for themes

During this phase the intention is to re-focus the analysis at the broader level of themes rather than codes, which is described by Braun & Clarke (2006) as sorting the different codes into potential themes. We started working individually, placing the codes into broader themes that could be useful visualising the contours of the story we wanted to tell through our data material. We found it helpful to collaborate on finding potential themes and then continue to collect relevant coding within the identified themes individually (see Picture 2). We also compared and discussed our work and how to proceed.

As argued by Thompson (2022) thematic analysis is an increasingly popular method to analyse qualitative data that captures patterns across the raw data and structures the data into meaningful themes, and is presented as a flexible technique that can be employed both within inductive and deductive research designs (Thompson, 2022, p. 1411). As described by Braun & Clarke (2006) themes or patterns can be identified in one or two primary ways in thematic analysis. There is an inductive “bottom up” way or in an deductive “top down” way (Braun & Clarke, 2006, p. 14). A third option within research design is abduction, which aims to find a middle ground between inductive and deductive methods (Thompson, 2022, p. 1411). While an inductive coding process would develop themes from the data, an abductive coding process combines inductive theme development from data with deductive themes from theory. Benefits of abductive coding is that it both identifies themes from the data material, and informs these themes from theory, contributing to theory development (Thompson, 2022, p. 1411).

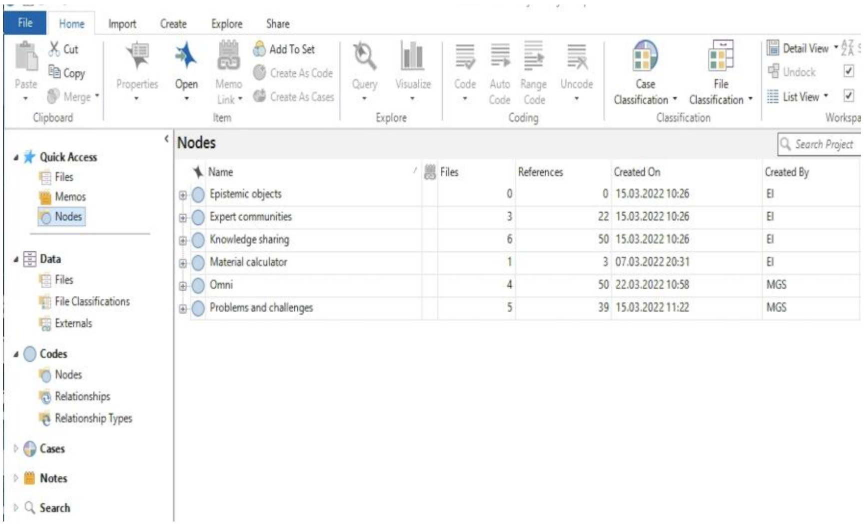

Nodes

Name	Files	References	Created On	Created By
Epistemic objects	0	0	15.03.2022 10:26	EI
Coding and Delfi	2	11	06.04.2022 11:56	MGS
Drivers	1	2	06.04.2022 11:56	MGS
Functions	0	0	06.04.2022 11:57	MGS
Sales	1	3	06.04.2022 13:52	MGS
Software development	2	12	06.04.2022 11:56	MGS
Expert communities	3	22	15.03.2022 10:26	EI
Areas of expertise	3	14	06.04.2022 11:57	MGS
Coordinating expertise	3	8	06.04.2022 11:57	MGS
tasks and responsibilities	4	20	06.04.2022 11:57	MGS
Knowledge sharing	6	50	15.03.2022 10:26	EI
Customer Support	5	10	06.04.2022 11:58	MGS
Dependencies	1	8	06.04.2022 11:58	MGS
Documentation practices	2	11	06.04.2022 11:58	MGS
Knowledge sharing proctices	6	18	06.04.2022 11:59	MGS
Student connection	4	11	06.04.2022 11:50	MGS
Weekly meetings	4	10	06.04.2022 11:59	MGS

Picture 2: Searching for themes

Reviewing themes

During this phase it became evident that some themes didn’t serve the purpose of being themes the way we initially thought. As described by Braun & Clarke (2006), there could be various reasons for why a theme does not work out as expected. In our case it was partly because there was not always enough data material to support the themes, or we discovered that the data was too diverse. As presented in Braun and Clarke's (2006) paper on how to conduct a thematic analysis, there is a dual criteria for judging categories, saying that there should be an internal homogeneity and external heterogeneity, meaning that data within the themes should cohere meaningfully at the same time as they clearly differs from other themes. The main purpose of this phase is to review and refine relevant themes according to two levels, where the first one refers to reviewing themes based on the coded data extract, and the second level refers to reviewing themes in context of the entire dataset (Braun & Clarke, 2006, p. 20). We reviewed our themes accordingly to level one and level two and evaluated whether the associated data material was sufficient to preserve the relevant themes. From our research problem and theoretical framework, we identified four main themes from our data material. These were epistemic objects, expert communities, knowledge sharing and challenges (see Picture 3).



Picture 3: Reviewing themes

Defining and naming themes

During this phase, we conducted the final refinement of the themes with the aim of identifying the essence of each theme, and potential sub-themes that could provide relevant information to our analysis. At this stage of the thematic analysis we had already identified four main themes, presented in phase two. But we also identified more themes that proved relevant after reviewing the themes in the previous phases of the analysis. New relevant themes that were identified were sub-themes that could provide useful information about the two objects we decided to study further and follow up on.

Producing the report

The final step in thematic analysis, as it is presented by Braun & Clarke (2006) is to produce a written report in response to our research question. The purpose of the report is to tell the story of our data in a way which convinces the reader of the merit and validity of our analysis

(Braun & Clarke, 2006 p. 25). The thematic analysis is documented in our analysis chapter and discussion chapter. We have presented our empirical work in accordance with the structure we have used to present our theoretical framework so that the analysis and discussion appear coherent and logical.

4.5 Ethical considerations

Ethical considerations and legal guidelines are described by Thagaard (2018) as essential in all qualitative studies, and that this is because interviews conducted in qualitative research involves a direct contact between researcher and informant (Thagaard, 2018). Silverman describes that because qualitative research inevitably involves contact with human subjects in the field, ethical problems are a key issue (Silverman, 2017, p. 55). Therefore, researchers have a responsibility to ensure that the research takes place in accordance with recognized research ethics norms at all stages of the research as they are directly related to the integrity of the research and the disciplines involved (Bryman, 2016).

Informed consent is presented by Bryman (2016) a key ethical guideline that aims to describe what participation in the project entails, and the research participant should be given as much information as might be needed to make an informed decision on whether or not they wish to participate (Bryman, 2016, p. 129). Prior to the data collection we sent out an inquiry to all the participants where we thoroughly explained the purpose of our project, and that we planned to collect data through interviews and observations (see Appendix B). We provided information on why they were asked to participate in our project and what participation would imply for them. We also informed that it was voluntary, and if they choose to participate, they could at any time withdraw their consent without any particular reason. We thoroughly explained that they would not be identified in the thesis, that the data is securely stored and that we would process the information confidentially and in accordance with the data protection regulations, the General Data Protection Regulation and Personal Data Act (GDPR).

With the purpose of maintaining good research ethics, we sent the participants a consent form together with the inquiry that we collected written from each participant through mail (see Appendix C). Our understanding throughout the project has been that the participants have been positive regarding their participation and that they have been willing to share their experiences and contribute to increase our understanding of their practices.

As it is explained by Thagaard (2018) a researcher must develop an ethical awareness and it is essential that we treat our informants with respect and protect their autonomy, co-determination and integrity (Thagaard, 2018). During our interviews we considered it important to create a trusting atmosphere to make the informants feel safe and comfortable answering our questions. Given that the relationship between researcher and participant often is considered asymmetrical (Thagaard, 2018), we were concerned with appearing credible and leading the interviews in a reassuring manner.

We have collected, processed and analysed the data material in line with research ethical guidelines, and we have done our utmost to handle the participants with respect and maintain a sincere and honest approach throughout the project. In this context we consider that the ethical guidelines have been followed in our research project and appears in line with guidelines from NSD (Norwegian Centre for Research Data)

4.6 Credibility / The quality of the study

In qualitative research the terms reliability and validity are normally used to consider the quality of the study (Tjora, 2017, p. 231). The concept of *reliability* is connected to the question of whether a critical evaluation of the project gives the impression that the research has been carried out in a reliable and trustworthy manner and refers also to the question of whether another researcher will find the same results using the same methods (Thagaard, 2018, p. 187). To ensure the reliability our aim throughout the process has been to appear transparent and open regarding our choices and assessments. We have maintained a focus on keeping an open dialog with each other both through writing and conversations. We have kept a research diary where we have noted everything we did from meeting to meeting, and also what we planned to do the following days or weeks to ensure steady progress and transparent work. We used a recorder during all interviews to capture the participants' descriptions so that we could reproduce them correctly and ensure reliable information through transcribing all interviews. We have followed the group over some time and had two rounds of interviews and six observations. We conducted both digital and physical observations and it is possible to argue that this has had an impact on the collected data material, as the dynamic may change depending on our physical or digital presence.

The *validity* of the study is related to the results of the research and how we interpret our data, and the validity of the study is strengthened through critically reviewing the analysis process, and questioning our interpretations (Thagaard, 2018, p. 189). We have collaborated on analysing and interpreting our data, and we believe that we have benefitted from having two sets of eyes on the collected data material as it turned out to be a large amount of interviews and observation material. We also believe that being two researchers may have contributed positively in maintaining accuracy in terms of the content in the data material, as well as it helped clarify potential ambiguities during the process. We have grounded our analysis in the data material through thematic analysis and actively used quotes from a variety of informants and interviews to increase the transparency and validity of our analysis. We have supplemented interview data with observations and individual interviews with group interviews focusing on the same themes to strengthen interpretations and allow for participants to identify disagreements and correct any misinterpretations. We also considered it appropriate to preserve the team's work language so that their descriptions and interpretations were not lost in translations.

4.7 Summary

In this chapter we have presented our research design and methodology and approach to the data collection with regards to the broader research question of this study. We have done so by providing descriptions and justifications of our reflections and assessments with the purpose of presenting our work as transparent and credible. We have to the best of our ability conducted the study according to research ethical guidelines, and analyses of the data material and processing of information regarding research participants have been handled with thorough caution and only applied according to research ethical guidelines. We have done our utmost to be transparent about the process of planning and implementing our research study and achieve reliability and validity in our research.

5. Analysis

5.1 Introduction

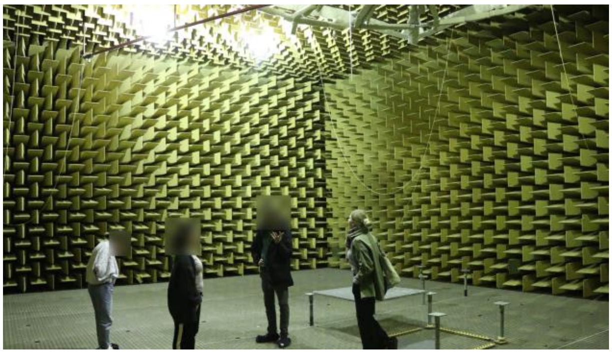
In this chapter we will present and analyse our collected data. We begin by introducing the company that we have chosen as our case and provide a descriptive representation of them as an expert community with a high level of specialised knowledge. Further we divide the analysis chapter into five subchapters. In chapter 5.2 we give a presentation of how the team worked as an expert community and how the team worked with several partial objects to facilitate knowledge development. Chapter 5.3 deals with epistemic practices and analyses how the experts engage in processes of exploration and materialisation. In chapter 5.4 we present findings on how the team shares knowledge through working with epistemic objects and how knowledge sharing was essential in these practices. Finally, in chapter 5.5 we present and discuss some key challenges the experts faced when working on developing the software, and how they dealt with the complexity and uncertainty involved. At the very end in chapter 5.6 we provide a summary and proceed to the discussion and conclusion of our thesis.

5.1.1 The case company

Our case company produced software for predicting room acoustics and PA-systems in large-scale buildings and areas such as concert halls, sports stadiums, airport terminals and outdoor areas with complex geometry distributed world-wide [webpage address not included to protect the anonymity of the participants]. The company was an active part of a university environment, and their offices were located close to the acoustics facilities of the university. These facilities included audio visual immersion labs (see Picture 4) and one of the world's most

Picture 4: Audio visual immersion lab of the university

advanced anechoic rooms (see Picture 5) where the team did measurements and simulations for developing and testing new and existing functions of the software. The company provided free software access for students and staff for educational purposes in exchange for access to the acoustics facilities of the university. The company also provided workshops and courses specially adapted to their customers and offered comprehensive software training combining demonstrations, hands-on exercises and challenging cases brought by the users. In addition, the company had a board which was composed of four representatives from consulting companies, who were also users of the software themselves and contributed with input on new features and plans for development of the software.



Picture 5: The anechoic chamber of the university

The team comprised eight members with a high level of expertise relevant to our research problem within the fields of acoustics, mathematical modelling and computing. The way the team was organised indicated a large degree of flexibility and autonomy, regarding work hours and location. Most of the team members worked on individual projects such as new functionalities for the software or production of theoretical papers for presentation at conferences, but usually shared their progress and discussed potential challenges with the rest of their team regularly, and it appeared to us as the company had a clear open-door policy which made the team available to each other.

The company had four offices where the founder, the manager and the sales- and marketing secretary each had their own office, whilst the rest of the team shared an open office space with desks and a meeting table (see Picture 6).

Picture 6: The team working in the shared office space

A weekly meeting was held every Wednesday, and all team members attended either physically or digitally. The purpose of the weekly meeting was to update the rest of the team on each other's work and solve and discuss issues related to the software or potential challenges. Although the meeting was led by the manager, the overall structure appeared to us as non-hierarchical and flat, where every member owned their own projects and shared their progress with the team. In addition to managing the team, the manager acted as a coordinator who integrated new developments into the software and developed parts himself.

5.2 Working as an expert community

The team we used as a case to study our research problem consisted of members from six different countries world-wide. It emerged from the interviews that the team considered themselves a multi-disciplinary team due to their compound educational background from various branches of engineering, as well as physics and mathematics. All had their master's or PhD background within acoustics or mathematical modelling from the university where the company was located. Acoustical engineering was identified as the core disciplinary identity among the team members, even by the non-acousticians. Only one team member identified software development as their area of expertise. At the same time acoustics was not seen as a homogenous field of expertise: “We have a team that is very, how to say, multidisciplinary. They're plenty of them that work in acoustics, but I guess acoustic is also very wide and there are so many specialties in the team.“ (Luis, initial interview). Their common interests and professional identity united them in the work of developing the room acoustic software and contributed to us considering them as an expert community. It appeared to us that the development work had a strong scientific foundation in theory and research and was driven forward by the team's shared professional background and individual specialities, as well as their attention to active communication and collaboration:

> *We talk together on a regular basis every day…and some are good on the mathematics, and some are good on the physics […] there are different strongpoints […] But it is good to have that… you know, that everybody is not alike, and different*

things you are good at and different things you find motivating, so it's very good to have that in a team, and that is also where a team really starts to play out, when you have different strong sides and you recognise each other for it.
(Peter, initial interview)

The team dynamic was characterised by the facilitation of team members' strengths and specialties to positively contribute to the development of the software, and they appeared to have a large degree of autonomy where the team managed their everyday work tasks and work hours independently. According to the interviews, the manager played an overall coordinating role and was usually the one who initiated new development projects, as well as worked with the integration of the developed projects. From our observations and interviews, he was the one who prioritised, made final decisions, coordinated and followed up tasks and projects. Even though he as the manager would have the final word, he left a lot of creative freedom and responsibility for the developers to further develop the projects. From their own descriptions the company had a casual and relaxed atmosphere and way of communication that gave the experience of a flat hierarchy. At the same time they respected the knowledge and experience of those who had been part of the company the longest. Although the team was an expert community who mostly shared the same technical background in acoustics, they still contributed to the team with their individual strengths and resources.

Many of us at the company have like…, we don't have like a specific task, but we're kind of multitasking and we kind of... imagine like a Venn diagram with plenty of circles, then that's kind of, like how the company works. (Luis, initial interview)

This metaphor of the team explains how the members had both intersecting and separate skills, being part of a whole and at the same time having their distinctive qualities.

5.2.1 The expert community and object work

The primary epistemic object of our case is a room acoustics simulation software. The software was one of the first on the market and was developed at the university over a period of seven years before being launched in the 1990's. It was developed with research funding and co-funded by consulting companies who were themselves potential users of the software and willing to invest in its development. The software consists of various tools or functions related to auralisations, simulations and results of measurements, which are the main outputs of the software. While each function has a specific acoustic-related purpose in the software, we are mostly interested in how the experts work on developing the functions.

The work varied from handling short tasks that can be quickly solved, to long-term projects that required a lot of resources and collaboration. From our observations, continuous efforts were made to match the tasks with the employees in the team, depending on their expertise, skills and competences. They all communicated a shared understanding of the importance of delegation of tasks. Delegation took place particularly at weekly meetings, as well as through the task managing system that assisted in this process. The team also had a shared responsibility and division of labour in managing the technical support function, where they assisted their customers and users if they experienced issues or bugs in the software.

As the team described their way of working, different work practices unfolded. How each team member preferred to work was closely related to the tasks they were to perform and each followed their own individual patterns and schedules. Some preferred being in the office where they could talk directly to a colleague discussing a potential issue or challenge, while others experienced that their tasks were better solved at home. Regarding research and development, the work was described as a dynamic process where "the task is to come up with the tasks" (Victor, group interview). The process was driven forward by discussing and solving problems that came up in the development process which added new layers of complexity and variation to the software and enabled the development of new functions. Development problems were described in interviews as being blurry and untransparent, full of challenges and uncertainties and the experts were in a continuous process of figuring things out. Some challenges were described by the team when it came to working with the research and development part where the process was perceived as uncertain and the tasks were not necessarily clearly defined, meaning that a lot of their work was about coming up with ideas for new improvements and developments.

> *It's a bit of a challenge, but it's an exciting one. And when… I think when you are in Research and Development you have to face that anyways (…) That's the nature of development.* (Victor, group interview)

The challenges the expert community raised came from working with epistemic objects.

5.2.2 The expert community and epistemic objects

From the interviews, two characteristics were prominent in the expert communities' work when developing the software, the *functions* that they intended to develop and the *drivers* that triggered their development:

> *So then we try first to identify what we can work on, and what we can improve in the software, and that comes from different things, that can come from our own ideas, it can come from customers, the customers are really open to giving us suggestions, the manager's ideas as well, the board, so then yeah, when we have these ideas, we kind of discuss what makes sense for me to work on and then.* (Victor, initial interview)

The *functions* were described as either development of new functions or tools in the software or improvements and maintenance of existing ones. New functions were also reported to be developed based on the knowledge in existing functions. An expressed ideal for the software and its functionality was that it should be experienced as both useful and user friendly by the end user. The user interface and user needs were thus central considerations in the software development. Behind the functions of the software were complex knowledge such as models, formulas and algorithms that were coded into the software. A core task of the software was to make complex knowledge into visual (simulations) and audible (auralisations) representations that were simple to use by the end users:

> *I think one of the highlights of [the software] is the whole graphical user interface, and how easy it is to like setup functions and run calculations, because there are*

other software that are not, maybe not exactly like [the software], but they are used like for similar things. (Luis, group interview)

Reaching a version ready for launch and to be used by customers involved a balancing act between the level of detail and necessary simplifications, between quality and speed:

> *…we try to use kind of practical or engineering thinking. So that is in contrast to having a what should I say, a purely hundred percent, perfect idea, thinking that we will solve this mathematically correctly and will not deviate from the absolute perfect solution. Then our calculation would never finish. So it's important that we find the balance between what will work sufficiently well. And yeah, so that is kind of the main thinking in the development of the method that we should not necessary go for the perfect solution, but it should be as close to perfect as possible, but still within the practical limits.* (Jacob, initial interview).

If the software functions involved a too high level of detail, it required too much computer power making the software slow and hard to use. On the other hand, if the calculations were too simplified, the results became less accurate meaning the output was less reliable. To avoid these a compromise was made between the two when working on developing the software and its functions.

Deciding what function to develop was described by the experts as a dynamic team process triggered by different drivers in the software development. These *drivers* were both external and internal to the team. *External drivers* were drivers mentioned in the interviews related to developments in the wider field of expertise, and to the stakeholders that used the software and had a channel where they could give input. The *internal drivers* mentioned were related to the inner discussions and dynamic processes of the team leading to the decision of what to develop next. We start with the external drivers and work our way inwards. The developments from the wider field of expertise were driven by professional, technological or theoretical concerns.

Technological developments related to what technology existed and what was considered as technologically possible:

> *Computers change. There will, we always need to follow closely the development in the hardware and development in software, and it is important to keep up to date.*

> *Because, well, if there's a new version of the windows coming up or whatever, then we need to test it as quickly as possible. To make sure that we don't face any new problems. So this is one challenge to follow, that what happens in the technology outside of our own company.* (Jacob, initial interview)

As the quote shows, this development could be pushed from the outside. The expert community also mentioned professional developments, including new standards and practices of sound acoustics such as building models standards and international sound standards, as well as acoustic databases.

Theoretical developments came from research and the development of acoustic theories. Here the team of experts described how they both contributed to the development and collected knowledge by publishing and reading scientific papers. An example was the company's practice of attending international acoustical conferences where they presented papers based on their own research and development processes, and preparations for these conferences were often discussed in the weekly meetings we observed. The team explained how conferences were dynamic arenas of knowledge sharing where theory could be shared and discussed, and the company got feedback on the software and the presentation of papers:

> *Through conferences there is some knowledge sharing that you can see what others have done. Either because we have been there so we see some presentations. Also because we have an exhibition booth usually and then we get people coming, stepping by discussing with us about the software and new developments. Also because actually almost every time when a conference is finished we get to see the proceedings, you know all the list of papers and try to see which are relevant for us. Maybe some of them use [the software] already or some of them are generally about acoustics. And that gives ideas of developing new things.* (Adam, initial interview)

Another important institution was the university, and there was a student assistant on the team working for the company while taking their own degree in acoustics, acting as a bridge between the company and the university. As part of their work the student assistant made measurements and tests that contributed to developing both university knowledge and the software. They attended weekly meetings and were highly integrated into the work of the company and received support and feedback from the expert community where other team members acted as supervisors.

Moving from the field of expert knowledge to the field of software development, competitors who produced similar software were also described in interviews as drivers for software development. The expert community was not stressed about competition, but rather saw them as inspiration to develop new functions:

> *… we can see for instance in a conference or in their ad if they have developed a new feature. And say, mm, yeah maybe that takes part of the competition, maybe we should also do something similar or these things. So competitors also you know give initiation for developing new features.* (Adam, initial interview)

From the interviews, the team of experts expressed little concern about competition in general because of the company policy of being open (see chapter 5.4). Competition was seen as a driver for developing new functions and solutions to knowledge problems based on their openly available functions and documentation.

Customers were seen as being in the centre of developments: “It's very much the users that push the development and it has been so for decades I think” (Jacob, group interview). From the interviews and observations, we could identify several ways in which they drove software development. First, they asked for a functioning user interface which was an important consideration since the beginning: “I recall that was an extra part in that there was industrial partners in it and they kind of asked for a user interface. So that was a big focus that it had a user interface.” (Peter, group interview). Second, they expressed needs coming from their use of the software and asked for new functionalities to be developed:

> *Eh, a very big percentage of the support conversation ends up in improving the software. Either because there is actually a bug. The customer says “there's something wrong with my calculation, what's going on?”. Sometimes it's their fault because they didn't use the procedure correctly. Sometimes it's because we have a bug. We just released a new version, something has changed and so on that helps us improving the software. Or some other times we figure out that what they do could be done in a better way and then we initiate the conversation for a new feature. A new tool. Sometimes they come right away and say hey guys I want this tool.* (Adam, initial interview)

Third, they asked for support and gave feedback on the existing software and its functionalities which could lead to problem solving, improvements and new functionalities.

This was part of a deliberate strategy: "We let people use it. We let people see where it performs well and not. And according to that, then we will see what we can do" (Victor, follow-up interview). It was then up to the team to translate these needs and requests and decide if they lead to changes in the software. Beside customers, another stakeholder was the board of the company. According to the team, the main drivers of the board were not financial: "the board is only interested in having the best possible tool because they use it themselves in their own company" (Jacob, group interview). They contributed by expressing needs they had from the software and also gave input to the expert community. What function to take on based on the different drivers was decided by the team in internal discussions.

The internal discussions were part of the dynamic process of selecting functions to develop. The company had a long list of potential functions that could be developed:

> *I have a long list and then it's mainly me that compiles that list of things that could come in, but then we talk about it and we develop sort the idea of what it could be, we do go to conferences and sometimes there is a conference paper that came with something that could be an inspiration and probably it is not going to look at all like that when we put it in the software in the end […] It is a dynamic process so once you kind of develop one thing you also learn something new, so it would be, the answer to the second problem or third problem wouldn't be the same after you worked on for a year, and you might also realised that, that that project over there is undoable, it is never going to work, people might not be interested in it after all.* (Peter, initial interview)

The team described several factors that influenced the selection. First, the usefulness of the function and if it would improve the usability of the software, as the team raised usability as an important feature in the group interview. Second, to push the limits of what was technically possible for the software to do acted as a motivation and a driver: "very often I just for curiosity test the program to see what way are the limits and especially when the new version is getting ready to be launched." (Jacob, initial interview). Third, fixing "bugs" in the software could trigger discussions about new ideas and improvements of existing functions, as we observed when discussing support in weekly meetings. Challenges identified in the software lead to reflections on how to integrate and combine existing functions in a better way.

Discussions evolved around sharing ideas: "For like coming up with ideas, actually talking them through, talking it through is the best, I think. So, I really enjoy talking with [Daniel, Peter or Jacob]…" (Victor, group interview). Coming up with ideas and then sharing them with the team for discussion could thus lead to new functions being developed. Sometimes a shortcoming in the software was the result of being dependent on a third-party solution that did not work optimally, and a new function was developed based on the idea that the company could do it better themselves: "*Jacob*: Yes, so we decided it was better to do it ourself. *Peter*: To build our own." (group interview). Based on these different considerations the team would discuss what part of the software should be the focus of improvement and development, also bringing into consideration the current expertise available on the team: "there are different persons of each of us that are, have a strength in one area. Both before coming here and also if you work more with one thing then you learn more about it" (Peter, group interview). While the manager had a final word, he was also dependent on the inner motivation of the team members and the function had to be matched with the team member who had the optimal expertise to start the knowledge development process.

5.2.3 The Omni source and the material calculator

From the data material there were particularly two objects that appeared especially interesting and relevant to describe the teams' practises, and which we have chosen as actual and current examples of partial epistemic objects. The Omni source is a special loudspeaker used for measurements and a stand-alone hardware product. The material calculator is a tool developed to be used in the software. In this chapter we will describe these two objects and in later chapters use them as examples of epistemic objects, epistemic practices and knowledge sharing.

The Omni source is a special loudspeaker for measurements. The object is a speaker, while *source* refers to its function as a sound source used in measurements. It is a stand-alone hardware product with the purpose of making the sound measurement process easier. The speaker is a lightweight portable device consisting of a loudspeaker unit with a *doughnut-like* lens with electronics and batteries on top and with a hole in it to enhance high-frequency omni directionality, meaning that the loudspeaker transmitted sound in all directions. This loudspeaker is the only hardware product the company has produced and an improvement from earlier solutions by competitors.

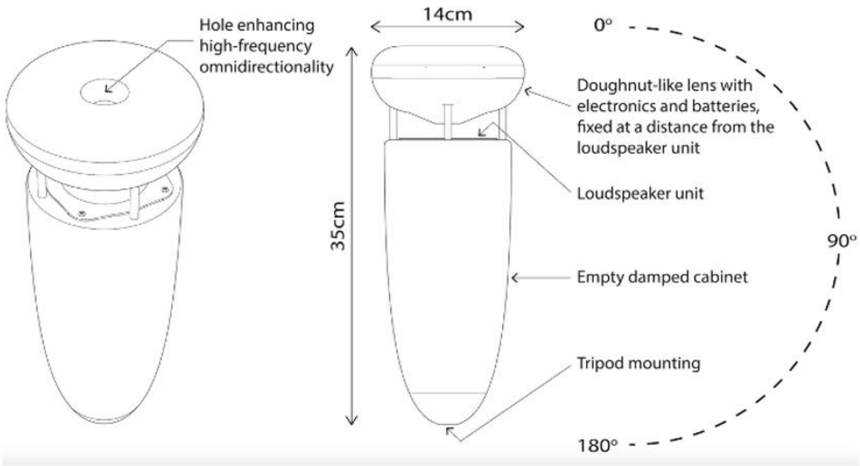
The material calculator is a tool developed to be used in the software. It was made from scratch as a stand-alone function and later integrated into the software. The software allowed the user to calculate how a specific material would reflect sound and the results could then be imported into a room model and used to project simulations:

> *Basically what you do in there is that you build a stack of layers, all of them having a different acoustic properties and then the material calculator will like, yeah, well calculate how it reflects and absorbs sound.* (Victor, follow-up interview)

It had different parameters that could be manipulated to match the materials used in the room model and it was a part of the input data in the workflow of the computer simulation software: “what this tool offers is basically the whole train of thought until you get to the simplified result” (Victor, follow-up interview). The material calculator was developed based on several acoustic theories, formulas and models that were integrated into the function.

5.2.4 The Omni source as an epistemic object

The team developed the Omni source with the purpose of solving problems in room acoustic measurements. It consists of a single loudspeaker unit, which keeps it small and close to what the developers describe as an ideal point source (see Picture 7). The loudspeaker delivers an omni-directional sound radiation pattern, with advanced performance at low frequency, while being fully portable and wireless.



Picture 7: Sketch of the Omni source

It was described in one of the follow-up interviews that there have been several measurement sources available within the acoustical field, and that it has been possible to measure room acoustics using different sources, although this has caused challenges due to the fact that the measurements were not standardised:

> *So, you could go and measure room acoustics with any kind of source, but then it would be a different source every time you had an acoustician going and measuring. They are all different. […] So that's not what we wanted from the measurements speaker for room acoustics. So therefore, we decided to make it sort of omni directional, radiating sound in all directions.* (Peter, follow-up interview)

Due to this the results would vary given the quality of the sources. One of the developers explained in their follow up interview that some sources had sufficient directionality which was appropriate for hi-fi but were perhaps designed to make adequate radiation directly towards one specific point in a room and were therefore not able to capture the room as a whole. This was not sufficient from what the developers wanted from a measurement speaker and can be perceived as an *internal driver* that led the team into discussing the possibility of developing a speaker for room acoustics that were omni-directional, radiating sound in all directions, and at the same time emanating from the needs of end users.

It became clear from the follow-up interviews that it was the manager who first initiated the development of the Omni source due what they described as years of frustration related to problems with room acoustic measurements, both practical and acoustical. The developers initially made a version with two speakers against each other that could make an omni-directional pattern, unfortunately this solution did not give an optimal frequency distribution due to sound distortion. As the development process continued more questions were being generated regarding the design and the complexity in combining the acoustical requirements with the ambitions of keeping the device wireless and portable:

> *But the most important thing was once we found the shape, what do we do with all the rest? Because we have batteries and we have circuits [...] My big idea in this project was to decide to put everything, or all the necessary equipment in the lens [...] I said, since we have that volume on top and we don't really use it for anything else. Why don't we use it to stuff everything we can in terms of equipment and batteries and so on, and then I started working that way.* (Adam, follow-up interview)

In the process of trying to make the loudspeaker into a concrete hardware unit the developers explained in their interviews how they started making models of the speaker in an external software, experimenting with potential designs. Eventually they continued with 3D-printed prototypes and tested various shapes and ways of putting them together (see Picture 8). Once they had identified an ideal shape the main developer came up with the idea of putting all the necessary equipment inside the lens on top of the speaker unit, rather than in the base of the speaker.

Picture 8: 3D-printed prototypes for the Omni source

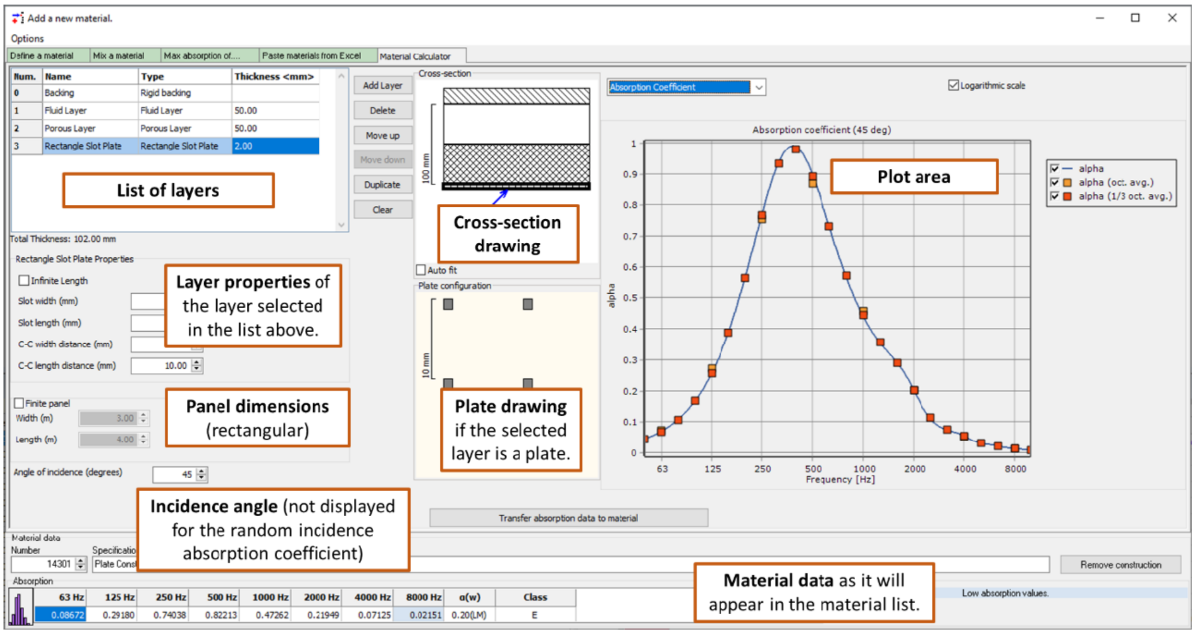
5.2.5 The material calculator as an epistemic object

From follow-up interviews with the main developers, we gained insight into the complexity of the material calculator and how it was developed. The material calculator combined several layers of material properties each made up of different theoretical models, in order to produce a final output. The idea of the tool came from the main developer who felt a great ownership to the project:

> *I guess, that I feel it's a bit my baby. The way as I said it is like. I think one of the most rewarding things of the, like all the programming thing, when you actually build something and it works and people like it and people use it. So that's very nice.* (Victor, follow-up interview)

The tool built on the existing software and research on material properties coming from amongst others "Victors" acoustics PhD and research of the senior researcher. It could perform multiple functions and had been developed from scratch as a stand-alone tool using Delphi integrated development environment, the same development environment used to develop the rest of the software. The tool was developed separately and later integrated into the main software. Developing it from scratch gave the main developer a unique understanding: "it was also for me a way to understand how the whole architecture works ... which was very interesting" (Victor, follow-up interview) and he described the process as a great learning experience involving several rounds of collecting, reading and discussing expert knowledge. The main function was to calculate how materials reflect and absorb sound, but the tool also had several new experimental functions. In the tool the user could compile different materials in layers, the order and properties could then be manipulated by changing different parameters:

> *So if you take this particular example, we start with a rigid termination so that can be a wall. Something hard, then here we have a cavity.[...]. So we make this little illustration of it. And then for each of these layers you can kind of change some parameters. And see how it changes, the absorption coefficient. So how sound is absorbed for things.* (Victor, follow-up interview)



Picture 9: Material calculator user interface

When parameters were manipulated the results became visible immediately in a visual way as graphs which made it easy to see the effect changing the parameters would have (see Picture 9). The interface of the calculator was purposefully kept simple to enhance usability. The visual part of the tool gave it an educational potential:

> *So as you can see we try to make it quite graphical, with a lot of information as well that people can use. And I think what is really valuable as well with it is how, how do you say... how you can use it to understand what's going on, like, as in a bit of an educational tool like. [...] And it's something that I would have even liked I think in my studies too, because I remember saying, okay, if you put a cavity, then the absorption coefficient will behave that way, but it was very abstract. And this makes it very illustrative the way so you can sort of play with it yourself in a way, which is quite nice.* (Victor, follow-up interview)

The calculations from all the layers compiled could then be transferred back into the main software to be used for simulations and auralisations. The expert knowledge needed to be simplified to be applied: “The way these materials are modelled in a way. They go quite deep into the physics of stuff and then to bring it back into [the software] there's a lot of simplification steps.” (Victor, follow-up interview), and the developers described facing several challenges when trying to get the different types of layers to behave as they wanted:

“You know that [senior researcher] was speaking about the membrane that still doesn't really behave as you would like to. And there's a lot on it as well.” (Victor, follow-up interview).

There were external drivers coming from the wider field of expertise from both international standards as well as scientific knowledge and theoretical models: “And that one, I had collected from the literature. Found in the library or where... I found models, equations, things that could be used to estimate sound absorption of different kind of materials, a lot of graphs, some simple equations.” (Jacob, follow-up interview). Standards were embedded in the software and updates in standards triggered new developments. Scientific knowledge constituted the foundation of the tool and was influenced by educational institutions, research communities and the work of the expert researchers in the company. The senior researcher drew on a long career of research and publications as a starting point for exploring the knowledge they needed to develop the tool: “this is a product by I also could draw on experience from my entire career“ (Jacob, follow-up interview). The main developer learned the basic acoustic models in his studies and had a relevant PhD on material properties, while the manager supported the process. They collected knowledge by inquiring into the wider field of research looking for relevant models that could be applied in the calculator.

The developers said they were inspired also by other similar tools. A customer who compared tools and got different results contacted support:

> *We want it to be transparent so that people know exactly what's going on. [...] One of our customers that wrote about... the tool and he was comparing with another tool. Ah, in this other tool they do it like this, can you explain? Well, in the manual of the other tool they say just that "ahh, we do this...". So like we try to be as explanatory as possible.* (Victor, follow-up interview)

To answer the questions, the team needed to get familiar with other software and tools to understand how it worked differently. When this led to identifying lacks in their own tools, it could generate ideas for improvements.

Customers also acted as a driver in the development of the tool: “We did have for example [...] customers, I don't know, like maybe three-four, sometimes that would write to us: "Hi, I'm having this material, but there's a cavity behind. What do I do?" (Victor, follow-up interview). Before its existence the company would get support requests regarding material properties and the team then had to manually calculate these:

We know these models are here, we studied it, you know, in our masters and stuff and yeah, and the customer demand as well in a way. … they may not have said specifically we want a material calculator, but the question made it clear that if this existed it would be very nice. (Victor, follow-up interview)

Indirectly the material calculator answered the needs of the customers, but it was the experts that needed to manage the wants of the customers and translate them into a tool that answered these needs. Deciding to develop a material calculator came from the idea being raised and discussed internally on the team.

The tool integrated several different aspects from the field of expertise, introducing a high degree of complexity to the object. Several new theories and ideas were put inside it at the same time: "There were two, three different aspects of it, that were new theories, new ideas that he had. But then everything was kind of added on top of it, you know, you cannot really separate the different aspects" (Peter, follow-up interview). For it to become a usable tool the complexity needed to be reduced, and some shortcuts and simplifications were necessary:

Of course, I made some shortcuts and some simplifications, but it was (inaudible) to get, well, a reasonably correct function for the angle dependency of the sound reflection, because [...] is quite complicated how the sound is reflected as a function of frequency. (Jacob, follow-up interview)

Not all could be accounted for in the first version of the tool: "You have a lot of things happening on the microscopic scale as well, which we haven't gone so much into that." (Victor, follow-up interview). At some point the tool needed to be realised into a functioning version that could be launched with the software and there was no time to solve all the issues that came up: "And then the idea is that [Victor] and I will continue to work with some of the more advanced ideas which we didn't have time enough for. But we thought what we had was enough for it to be useful at present state". (Jacob, follow-up interview). Being useful, they could finalise the tool for a launch of its first version that was distributed with the last edition of the software. The advanced ideas could be finalised and implemented in future versions. Even so, there was no clear plan of where to go next (see 5.2.2). When working with developing the calculator and source, the expert community engaged in epistemic practices.

5.3 Epistemic practices

In our interviews and observations examples were given of how the team of experts engage in software development practices when developing the software as an epistemic object. Only about half of the team said they engaged in coding practices regularly. None of the experts had a software developer background, even if this is central for developing the software. Some did mention having programming as a subject in their engineering studies, but even then, coding was a process of learning through trial and error where they researched solutions to problems as they appeared:

> *I was not educated as a software developer, so I kind of learn as I go, so its more , I don't know if what I do is the proper way, I don't know if there is a proper way actually, ehm, so its more, yeah I kind of go by common sense, I mean, not being a software developer its really, it always an acoustic idea behind, so its starts with for me a lot of drafts (shows his drafts) I like writing on paper a lot, so like to get ideas in there, and then I try to implement some steps, I found out that if the task is too big its actually not manageable, so I'm trying to cut it down in small milestones, and maybe I'm going to make this work first, and then this and this, and trying to test as I go, make it as I go.* (Victor, initial interview)

This reflects the team's engineer mentality of finding practical solutions to problems. The team expressed having an autodidact approach to coding, and they didn’t mention using any specific established methodology of software development.

The practice of software development took place in a tool called Delphi, an integrated development environment that allowed for programming directly into the code as well as using the drag-and-drop visual interface for simplifying the development process. They described it as a complex ecosystem:

> *Peter: So saying Delphi, Delphi is the compiler which sort of, you make a visual design of what you want and you write the code and it compiles it into the executed version.*
> *Adam: Programming environment.*
> *Victor: And also in Delphi we use a lot of libraries,*
> *Peter: Yeah*

Victor: Like third party libraries, then we can, like we don't code everything from scratch.
Interviewer: No, okay.
Peter: And there are some massive libraries for all the input boxes and graph... and if you want to have charting in your program and if you want to do signal processing.. resampling.
Victor: it's all about understanding them afterwards, but
Peter: yeah yeah yeah, and it's complex tools and sometimes not fully documented really. (laughs)
Luis: (laughs) oh yeah
Peter: Which is kind of again research, figuring out how the hell it works. Or suppose to work.
Interviewer: Yeah.
Peter: Eh, but there are many different... it's an ecosystem.
Interviewer: Yeah, that's interesting. An ecosystem.
Peter: Like many tools in it.
(Group interview)

Delphi helped structure the development process as different functions could be stored in units, a type of folder in the tool. It allowed for compilation of all the different functions in development including the use of third-party libraries and other tools, avoiding having to make everything from scratch. Coding practices could include the use of simple tools like notepad-software for comparing codes, as well as using task management systems to track different issues outside the code. Excel made it possible to make calculations outside the software development environment and then compare the results with the simulation results. The results of simulations could also be extracted from the software into excel files for further investigation.

The team described in interviews how they were in a constant process of improving the software. Versions used for development were kept separately from finalised versions published on online servers as software packages and updates. Multiple copies of the software acted as closed environments where they could work on developing different functions separately from the master version:

It's partly a manual process so that comes back to what I said, that I kind of the integrator in the end so I sit with the master version. So different employees develop different features and we do it a continuous sort of process where we work together and discuss it ... when it's finished I need to put it into the master version of it all, connect the rest of the buttons and, so give it a coherent kind of appearance and make sure it works. And then we have a new master version and we work on from that. (Peter, initial interview)

Some of the projects of developing functions were described as more short term, while others were long term development processes that could take up to a few years to complete. New releases were central to the development process and acted both as versions that needed to be finalised in order to be launched, and that needed to be monitored and followed-up after launch when bugs and other shortcomings became visible:

[The company] you have to think it's a company that produce software. And every year and a half, two years, there is a new release of the software. And that's an important moment. Meaning that the most of our effort is tested to somehow, put it out and resolve the problems that. Yeah often, when you have problems with something they appear very, very, very soon when you release some code because there are many users using it and they will tell you that this is wrong. (Daniel, initial interview)

Reports came from customer feedback which often led to patches or smaller updates being issued in between versions. When new versions were finalised, all the different parts that had been developed separately had to be integrated manually. An extra layer of complexity was added because of the interdependencies between the different parts in the code:

So, to be aware that if [manager] for some reason has to fix something small in a unit that I have worked. Like in a part of the code that I have worked myself, then I have it in my list so I'm aware which lines that have changed and then we try to not overlap our changes. You know if he has done some changes to the same file and I have them in the same file then we have to match these files carefully. And I try to keep tracking, I have no... note pads or word documents when I keep track of this one's. (Adam, initial interview)

Changes in different functions could affect the same part of the code and developers needed to keep track of any interdependencies and changes in the code to avoid malfunctions. The team described the process of developing the different functions and integrating them into a version that could be launched as processes involving multiple iterations, where the new functions had to be validated through repeated testing of different cases. This was complicated by the many different development paths in the software: "...there are multiple paths in the software. One of them is, for example, importing of geometry, which it's something that really has nothing to do with acoustics. You can later use that for acoustics, but it's more related with mathematics and geometry." (Daniel, initial interview). In the initial interview the manager described the development process as unpredictable where one didn't know in advance if a function would turn out to work in the end: "Many of the things are sort of things that develop over years and we start having an idea of where to move, talk about it, start doing some mock ups demonstrating if it will work or figuring out that it actually doesn't.". We will now analyse how epistemic practices were involved in developing the Omni source and material calculator.

5.3.1 Epistemic practices involved in developing the Omni source

In an interview with the main developer of the Omni source he explained how the development process began as a creative solution to an acoustic problem with the purpose of creating a better alternative to the existing measurement sources. An example of an epistemic practice from the interview was designing the object and its functions. Although there was one main developer who worked with the developing the interface, there were several experts involved throughout the development process who contributed to increase the complexity of the loudspeaker in terms of its design and functionality:

> *The project was frozen a bit because we were stuck, wondering how really to manufacture because we couldn't do all the electronics ourselves, and for all these small details like for example mounting and screwing we wanted some mechanical engineers to get involved […] So we found a Greek company who, you know, they are specialised in making so-called embedded system. So they are specialised in designing the circuit for our purpose and then designing all the mechanical components. Like, how do they screw these things together.* (Adam, follow-up interview)

The developers also explained that in addition to external experts who contributed to the design and functionality process, there were external participants involved in producing the associated casing, a backpack specially designed for carrying the loudspeaker together with a laptop (see Picture 10).

Picture 10: Presentation of the specially designed suitcase for the Omni source

In addition to his engineering education, we learned during our interviews that the main developer had a background in arts and was highly involved in the entire design process. Together with a former student assistant, he began to explore potential shapes of the loudspeaker through simulations in an external software. To find the optimal shape, the developers intuitively printed various prototypes and tested several alternatives by making measurements. The process was described as dynamic, and they adapted the prototypes and shapes as the process evolved.

> *We did some simulations in the very beginning, but eventually we started printed prototypes. […]we had already some good ideas how the main shape should look, that there should be a hole and so on so we said "okey, now we know the shape then we go measure it" and we didn't like it so much so we started going just by intuition […] So we're just by intuition changing our shape, printing it in 3D printers and then measuring again.* (Adam, follow-up interview)

The developer explained that the most important factor was to find the right shape, and a common challenge when designing a loudspeaker in general is that the speaker must have an empty cavity behind for it to resonate well. He further explained that it was challenging to find a solution where they could fit all the electronics and keep an empty cavity. As they came up with the idea of placing all the electronics in the lens on top of the speaker, new challenges unfolded as they had to find a working solution on how to fit all the electronics and batteries inside the lens without interfering with the acoustical properties, adding to the complexity of the object.

The design process also consisted of figuring out the best way to mount the lens onto the rest of the structure. The developers at the team handled and solved this challenge by trial and error based on intuition and knowledge, testing, and measurements. At the end of the process the manufacturer contributed by working out technical details in order for the source to be manufactured.

5.3.2 Epistemic practices involved in developing the material calculator

In follow-up interviews, knowledge validation was highlighted as a core epistemic practice used to solve problems in the material calculator. Collecting scientific knowledge would lead to many potential paths in the software and the knowledge needed to be validated in order for the software to give reliable projections, a process involving testing different models to discover which models and formulas were valid: "So we could see when they would work better or... which one was more simple, which one was wrong. So, yeah, that was kind of a lot of validation behind for sure." (Victor, follow-up interview).

They validated theoretical and mathematical models with data. This process could be challenging as there were several uncertainties coming from the theories themselves, which could be more or less accurate. It was described as a learning process:

> *So yeah, then you validate your tool with a lot of uncertainties sometimes, so this can feel a little bit uncomfortable, I would say. But you learn as you go and the more you understand the models, the easier it gets and the thing is that, yeah, in the end, yeah, this tool is pretty new. It just got launched. And yeah, we can still improve it.* (Victor, follow-up interview)

Uncertainties in the models could lead to uncertainties and shortcomings in the simulations. Models were validated by comparing simulation results with measurements in actual rooms, to identify the models that worked in reality: "So the, basically most of the theory has been made for infinite surfaces, but this never happens in real life. So then you kind of made up some, well made up... found some equations that kind of account for the fact that you have a finite size sample.” (Victor, follow-up interview).

The developers described finding discrepancies between the projections of the calculator and what actually happens in room measurements. Measurement data could also be unreliable and with errors, depending on how it was obtained, and having good measurement data to compare with was seen as crucial. Theories could also turn out to be inconsistent with data: “This one was from a very yeah, so kind of a combination of different old papers that seem to be quite complex. So probably quite accurate, but the problem is that this one, I never managed to fit it with like experimental data, or there were lots of questions.” (Victor, follow-up interview).

Different sound frequencies required different models, and models needed to solve the acoustical challenges of each of the layers. To find the best models they needed to understand the expert knowledge behind: “we would get results that were like, working better in some cases and worse in other cases and I was really trying to understand what was going on behind” (Victor, follow-up interview). The testing process was tedious, involving frequent testing and comparing calculation results with measurement data. The developers explained the process of testing of models as iterative: “So, this one is kind of an iterative process. We started with simple one, then we found the more advanced one and then an even more advanced one. So it went a bit step by step.” (Victor, follow-up interview). Both the senior researcher and main developer were involved, sharing documents with short summaries of new models they had tested: “Yeah, and then we have more tests and more tests and more tests. And it goes on. [...] Spend half a day or something, to, to do some tests and then report to [Victor] what I found” (Jacob, follow-up interview).

Getting the models from the article into something that could be tested and used, often involved extrapolating formulas from papers. The models were classified according to their theoretic underpinnings, if they were different or similar. Even after a long testing period, the calculator would not always behave as the developers wanted. It was a process of constantly trying to define the material calculator: “So we tried different things, but it was, yeah, in... I felt that a challenge was that there was no, you know, like... definite answer. Like okay, this seems to work better.” (Victor, follow-up interview).

Excel was also referred to as a tool commonly used for testing in engineering, as it gave a good overview and made it easy to locate errors by comparing calculations with the results of simulations, to find if the error was in the software or the theoretic model. It acted as a quality check on whether the model had been correctly coded into the software. Having the senior researcher on board was also an asset in the process as his experience could be used to quickly identify if the results were likely to be realistic based on past experience and a repertoire of expert knowledge: “He can just look at the curve and say yeah, this doesn't make sense. So this is really good (inaudible) to have.” (Victor, follow-up interview).

Even though the first version had been launched, developing the material calculator was an unfinished process. A main challenge was the lack of definite answers to the knowledge problem the material calculator represents. Two versions of the material calculator existed simultaneously, where one was stabilised into the latest release of the software, and the other version was for development. The developers kept an eye out for better models in the literature that could be implemented. There were no clear answers to the main knowledge problems: “We're not 100% sure. So this is always kind of the delicate part of it. So, yeah. Yeah, and then Yeah, I think that would be I would say the biggest challenges when I think about it.” (Victor, follow-up interview).

Epistemic practices can be seen as practices used to generate and share knowledge and the expert community used knowledge sharing actively in the practice of developing software. We now zoom in to analyse forms of knowledge sharing taking place in joint work with epistemic objects.

5.4 Knowledge sharing through working on objects

Different types of knowledge sharing activities took place. The categories below have been developed from themes appearing in the data and reflect different ways the expert community shared knowledge when working with joint epistemic objects. In summary they shared knowledge in software development through the activities of documentation, knowledge collection, communication, and collaboration (see Table 5).

Knowledge sharing category	Description	Examples from data
Documentation	A knowledge sharing activity involving creating documents with text, sound, images and graphs used to develop the software, train end users and share theoretic and technical findings.	Creating scientific papers and sharing them at international acoustical conferences.
Knowledge collection	A knowledge sharing activity where scientific and technical knowledge is collected, discussed and used to solve specific problems in developing the objects.	Reviewing knowledge from relevant research and development projects to use in development and support.
Communication	A knowledge sharing activity where the team shares knowledge by talking together, sharing updates, and giving feedback at planned and spontaneous meetings.	Continuously sharing information and feedback on each others' work.
Collaboration	A knowledge sharing activity of joint problem solving through frequent interactions and discussions based on interpersonal dependencies.	Generating ideas and solving problems through team collaboration.

Table 5: Categories of knowledge sharing activities

Documentation

Through the observations and interviews we identified documentation activities that involved making written, auditive, visual and graphical objects to share knowledge about the development and functions of the software. Knowledge was shared by creating and sharing different forms of documentation representing knowledge about the object. The documentation can be categorised according to three different central functions. The first was *documenting for developing* the software. Second was *making end user documentation* and involved sharing necessary information for customers to understand and use the software. And third was *making scientific papers* for sharing knowledge about the theoretical and professional significance of novel aspects of the software.

Documentation for developing the software was described in interviews as a documentation used by the people involved in research and development on the team. This activity of knowledge sharing was motivated by the need to know where things were and were meant to support development:

> *[...] you saw me trying to find this thing in the code too. It's not like, oh, I know what it is like, I kind of know where it is and you need to kind of go back into it and this can really help, knowing where things are, how they work. So that's what motivated me to take the time to write this.* (Victor, follow-up interview)

The documentation could either be written as metainformation in the code itself, explaining what that part of the code was about, or by making separate documentation as excel files, power-point presentations and documents. Writing comments in the code itself was common:

> *First I do it somehow in situ in the code itself I, very generous with the comments, only thing like that and I often put, for example, my initials in order to remember later that it was me the one saying that very bad thing or whatever, I mean, I see that all across the code it is not only me who is doing that and I'm probably I copy that way to yeah, [...] I make like a small introduction in the unit explaining what I'm doing and pulling some references.* (Daniel, initial interview)

The software code and development environment were seen as having knowledge built into it, amongst others it integrated and embedded models, formulas and other products of knowledge work: "you can say in these software development tools and component libraries and, eh, all this, it in itself of course shares a lot of knowledge, there's put a lot of knowledge into it." (Peter, initial interview).

As we saw examples of during interviews, separate documentation explained how the software was developed, the choices made, what different solutions that were tried out and what theory and expert knowledge was used. The separate documentation could be comprehensive documents or focused information about a specific issue. One argued for working more systematically with documentation: "Yeah, if it could be a good idea to do it more consistently, I think it could be a nice discussion to have because it makes it easier, I think to get into subject that you might not be working, might have not been involved with at the beginning." (Victor, follow-up interview).

We observed several examples of end user documentation like video tutorials and end user manuals that were shared openly on the website. Even if the target group was the end user, a lot of knowledge sharing took place in producing end documentation. Considerations were taken regarding the level of detail about the underlying expertise and how to present the function to end users without making it too theoretical. The knowledge needed to be translated so it was understandable:

> *Because, yeah, they sometimes they are very mathematical concepts that for sure the users that are aware of what I'm talking about. So maybe it's better to, I mean, I think I'm getting better into it and trying to don't get me into the detail in the things.* (Daniel, initial interview)

This describes how the knowledge sharing process involved the creator of the function, who contributed to developing a manuscript with examples of how the function worked. This was then subjected to feedback and revisions before videos and manual texts were produced. In this process "bugs", or errors in the function or code could also be discovered, leading to further development work taking place.

Scientific papers also served a role in documenting new functions and were presented at conferences or published in scientific journals (see Picture 11). The scientific papers shared theoretical knowledge together with results of tests and validation processes. This was a part of the company's policy of being open and sharing what they do, contributing to lower tension about competition:

> *Yes, I think. As long as we follow our politics, I'm relaxed. And the politics is we are open about what we do. We present our methods, our theories in the papers and conferences. Quite often we present the methods before they are actually implemented. We are open about what we do and we have high productivity to help our users.* (Jacob, initial interview)

Picture 11: Documentation - development of a theoretical paper on the Omni source.

Knowledge collection

When asked about how the team shares knowledge in the interviews, the informants described several *knowledge collection* activities where problems with the objects were solved by collecting, sharing and reviewing different knowledge resources such as scientific databases, online forums, google, etc. This could for instance happen when the knowledge of one expert or the knowledge on the team was insufficient to solve the problem:

> *And then it might be that nobody knows it internally and then we can start looking around for literature, and that's the normal thing in knowledge based sort of working, that we don't know it all, but [...] just the knowledge of knowing that this is this field [..] can be very useful knowledge for itself [...] just knowing the word to search for on google or whatever or in a journal or whatever can help a lot, or knowing that there is a person that we should talk with. [...] just having an idea of where to start looking for a solution to the problem, [...] most of the problems are like we don't know exactly eh, how to solve it up front right. If we knew we would have already developed it.* (Peter, initial interview)

Knowledge collection was mentioned to have happened when the team was faced with an unexpected error in the software. Customer support was a place where problems needed to be solved and discussing knowledge from external sources was involved:

> *You know, our users have possibility to ask for support [...] And normally it goes smoothly and it's not so complicated. But sometimes it is a tricky problem that is raised by the user. [...] and sometimes I interfere and go into my colleague who has answered say, well, we could add more to this reply. It is a little more complicated [...], take a look at this paper. [...] We can be a couple of persons involved in the discussion of the problem raised from the user. So I think answering support questions is actually a quite important part of our knowledge sharing.* (Jacob, initial interview)

In the interviews, customer support was described as a valued service for end users who depended on accessible and reliable support when facing software challenges. When a customer asked for support, a ticket was made in the customer support system. Some requests could be basic issues which were easy to solve, others were more complex software related knowledge problems that could lead to changes in the software. Whoever was responsible for customer support on that day received the ticket and if they were not knowledgeable about this aspect of the software, they asked a colleague in person or in writing, depending on their availability. The colleague then shared the specific knowledge or answered the support ticket themselves. In some cases more efforts were needed and the issue could be raised and discussed in a weekly meeting by the whole team who shared knowledge based on their expertise. If the issue could not be resolved with the knowledge of the team, knowledge and solutions could be searched for online.

In more complex support cases knowledge external to the team was needed. Knowledge could be collected by dropping questions in a specialised online forum, especially if the question involved a third-party resource, or by googling for solutions. When googling was part of knowledge collection, it was central to identify the right keyword by experimenting with different combinations. This was referred to by one of the informants as *discussing with google*:

> *Sometimes you can also just discuss it with google kind of (laughs), no, but searching for knowledge right, [...] it's not human but anyway you can kind of have a discussion with it in that way. You ask one question, you get an answer, you ask another question and then that... so the interaction, using tools and discussing with each other. … it is the development tool or it is sometimes google and sometimes articles and whatever, but the dynamic process that sort of throws at the wall and sort of see what comes back, [...] can help find solutions that you can't on your own.* (Peter, first interview)

Examples were also given of team members working together on searching and comparing solutions among each other in order to identify the best solutions. The activities of googling and discussing also took place when more theory or scientific knowledge was needed to develop new or improving existing functions, involving searching online research databases to find relevant research publications. External knowledge could then also be shared and discussed with other people on the team involved in the problem solving. The activity of searching for knowledge online was expressed to have speeded up the development of new functions substantially, compared to in the beginning of the company's existence when the internet was less developed.

Communication

A third category described in interviews and observed at meetings were that of *communication*, an integrated activity in the team. In interviews, the team reported that talking together a lot gave them a feeling of confidence and lessened worries and stress: “a typical day for us [..] starts with communication, and a good picture is that one day one of the leaders from the company on the above side. He came to me and he said, how can [Peter] always look so relaxed? (group laughs) And I said, I think it’s all about communication” (Rita, group interview). The dynamic activities of knowledge sharing involved internal communication on the team, as well as external communication with other actors and stakeholders. It took place at meetings, when talking together, and by giving feedback. Communication activities involving knowledge sharing were routinely performed at both planned and spontaneous meetings.

The weekly meetings we observed are good examples of routine communication: “Like every week we kind of touch base, what are you doing, what are you doing? So [Peter] kind of runs through all the things that goes on ...” (Rita, group interview). The weekly meetings were started to coordinate work after coronavirus restrictions limited the communication opportunities on the team. Everyone on the team was expected to participate at the weekly meetings, either physically or digitally (see Picture 12). Before weekly meetings existed, the daily manager coordinated the work by regularly communicating with each person on the team. The meetings followed a regular structure that the team was familiar with, but there was no official agenda for the meeting, giving it an informal character. The meeting took place in the largest office. Here a foldable table was placed with a computer screen on one end and a microphone in the middle of the table. The secretary would buy some cake or pastry and when people slowly arrived, they helped themselves with some cake and talked informally together. When all had arrived, the manager started the meeting by taking the word and introducing the first point of order.

Picture 12: The teams weekly meeting

From our observations we could see the informal character of the meetings. Even if the manager coordinated them, the team members took the word on their own initiative and jokes and comments were common during meetings. Weekly meetings were central to coordinating the team: "in the regular meeting we update ourselves and we kind of redistribute our tasks and remind each other "yes that has to be done"" (Adam, initial interview). This involved a high degree of knowledge sharing between the participants and helped the team to know what was going on: "basically in our [company] meeting, I mean, we talk every week about what the people is doing and yeah, we more or less know, even though we don't maybe understand the full details" (Daniel, initial interview). Some of the benefits expressed by the team of having the weekly meetings was that they gave comfort in the direction they were taking: "It's both to know that we are going in the right direction. For me it is hard to know that we are going in the right direction. Also, I would think that it's comforting for everybody to know that we kind of have an agreement of where we are on the way" (Peter, group interview).

We could observe how the team went through all main areas of work of the company, from customer related topics to the development of the software. Customer related topics covered sales, video tutorials, advertisements, homepage, courses and newsletters, while software developments covered scientific papers, conferences, support cases, new versions and follow-up of the last released version of the software, as well as main functions under development. While each of these topics were gone through one by one, there was some overlap between the topics and the conversation would move back and forth in a dynamic way. Knowledge sharing could take place at any of the topics, but some of the topics involved more discussions around knowledge problems than others. This was demonstrated in parts of the meeting dealing with development and follow-up of current and future releases of the software, as well as when discussing functions currently being developed. But also discussing support cases triggered processes of joint problem solving and knowledge sharing. This depended on the nature of the case, if it was merely routine or reflected a complex software issue. As with many of the conversations, they were not necessarily concluded, but several of the team members would give their contribution before moving on to the next topic. Knowledge sharing did not only take place during weekly meetings:

> *"Weekly meetings of course, that's a key point in this knowledge sharing. But outside these meetings. It can, it happens eh, from person to person when we are in the office,*

of course, and it happens through email, when we are.. in home office or remote." (Jacob, initial interview).

Of more need-instigated activities of knowledge sharing we could identify from the interviews the activities of talking together and giving each other feedback. Talking together was central to knowledge sharing as "most of the knowledge sharing we do mouth to mouth" (Peter, initial interview) and the team expressed an open and positive attitude to communication: "I guess I think have a good communication, so I never lagged, you know, communication with my colleague so... which is important. Like I never felt that I would interrupt somebody" (Rita, group interview). Being a small team, everyone knew each other and by talking a lot, everyone knew what was going on: "I mean that it's something that we talk all together. And what to do about something and then everyone knows what's going on and explaining that then [...] anyone can do" (Daniel, initial interview).

This positive experience of communication was also reflected in what can be described as a feedback culture on the team, where giving feedback on objects being worked on was an integrated knowledge sharing activity. There was a low threshold on the team to ask each other for feedback and giving feedback seemed commonplace:

> *I want some feedback from them in the beginning and then if somebody is here then I ask how about these subjects do you think they're nice and then I get an early feedback. Then I usually work myself for a while, presenting a final draft which then I sent to everybody and I want some feedback again. So I do ask for feedback often, as much as they do as well.* (Adam, initial interview)

Picture 13: Feedback from the team on the Omni source

They asked for feedback both in development processes and problem-solving situations (see Picture 13). Acts of giving feedback could take place immediately through just asking a colleague there and then, or by sending an email with a question or asking for feedback on some work produced:

> *The article we create that we also feedback and several rounds of feedback. Which comes mostly by email, but then. When it's a support question immediately or maybe there is a call, I don't know something. [Victor], do you remember that feature or that principle. Then there is like a immediate feedback and seeking of assistance.* (Adam, initial interview)

Feedback could also be given in several rounds when testing different potential solutions to knowledge problems. This would depend on the context. Short feedback could be given on specific issues, while regular feedback was common on joint projects. Even though the weekly coordinating meetings supported the team in sharing knowledge, this didn't replace the activity of updating and being updated by the coordinating manager. This was related to his role of integrating the work of the different experts on the team. Collaboration supplemented communication as a knowledge sharing activity.

Collaboration

Collaboration was identified as a knowledge sharing activity both in interviews and during observations at the company. Collaboration took place when two or more people on the team worked together to solve a problem or develop an idea by sharing knowledge. It involved direct interactions, discussing problems, and interpersonal dependencies between team members. Good communication and regular interactions were closely related as described in interviews, associated with little spatial and social distance:

> *And then, as we just talked about, also we have a very good communication and if I need something I just say "hey [Peter] can we talk about this?". So the distance is so short in order to get anything. If you have anything you need to discuss and then you guys are close to each other when you're here. So you can also just say «hey what do you think about this one?"* (Rita, group interview)

To be located in proximity from one another was an important contributor for collaboration to happen, and the team often turned to the colleague close to them or visited their office. There was a low threshold for contacting a colleague and asking for a meeting, and the manager's door was said to be always open if one needed to have a chat. As one team member said:

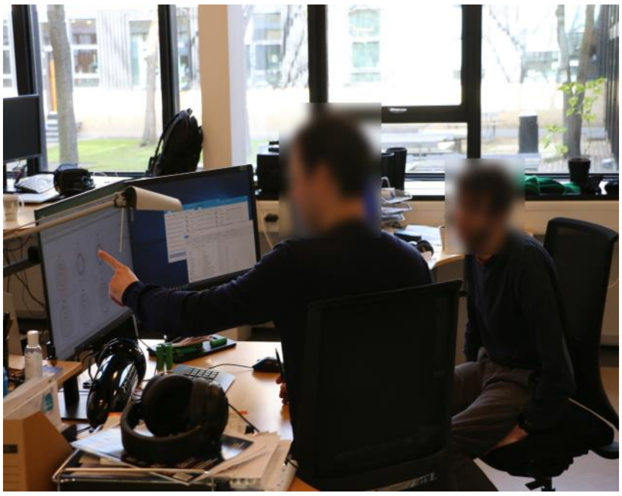
We work together in all kinds of connections because we are lucky enough to have a rather small team. So everyone knows the other one in the team. I find that very, very useful. [...] we know quite well each other and what are the strong and the weak parts of the profile. (Jacob, initial interview)

At the same time these daily interactions could also disrupt the daily work flow and lead to unpredictable and variegated work days: “So sometimes you kind of go to work planning that you will do this all day (group laughs) and then you don't touch it because this this this and that and there is support questions and there is questions on this one and that one” (Peter, group interview). Even if physical interactions were important, digital interactions supplemented these and took place regularly through sending direct messages, referring to each other the task manager system or writing an email. Interacting remotely were also commonplace on the team, especially during the Covid pandemic and some of the team members worked part-time or fully remote. Interactions would then take place as online video meetings where for instance screen sharing contributed to knowledge sharing by allowing two people to discuss the same piece of code together.

Discussing problems was a frequent activity on the team and associated with positive synergies and progression:

There will always be one person who is the main worker and [...] at least two persons involved in different aspects of development. [...] when two or more people are discussing a problem there is usually a synergy and you can step forward much better than if it's just a single person at tackling a problem. (Jacob, initial interview).

The team saw benefits of being more than one when developing the software: “when we do a merge of code and with [Peter] in the same computer, it's like there are four eyes looking at the problems that you can be doing, and then that's helpful” (Daniel, initial interview). By discussing problems the team reached agreement on what solution they thought to be the best to solve challenges in the development process.



Picture 14: Collaboration and problem solving, discussing graphics for the scientific paper on the Omni-source

The team depended on each other for problem solving and used each other to develop ideas and find solutions when working on development projects (see Picture 14). Dependencies were also reflected on how work was distributed, as the individual experts often were dependent on others doing their part and sharing their input in order for an object to be finished:

> *Currently there is some small in between project that my [...] colleague [...] is sitting with [...], meanwhile I am also developing something he needs to have before he can go on so I have some code where I understand that I need to implement somethings which he will be using and before that is then he can't really get on with this [function], [...] sometimes we have to wait for another one. The other trains need to arrive before you can change the tracks, right.* (Peter, initial interview)

The manager had a special focus on the user needs and software interface, as well as for integrating the different parts of the code into the final software version: “Actually, there is a big dependency when it comes to code on [Peter], who has everything on his yeah, then it's... in a way.” (Adam, group interview). In order to progress the team needed the manager to take an active role in facilitating knowledge sharing.

5.4.1 Knowledge sharing and the Omni source

The process of developing the Omni source comprised several activities of knowledge sharing. One of the most prominent was the activity of knowledge collection. The loudspeaker was the only hardware product the company had produced, making them dependent on different types of collaboration and methods to develop and complete the product, including external help from the manufacturer needed to produce the loudspeaker:

> *…I said “since we have that volume on top and we don't really use it for anything else, why don't we use it to stuff everything we can in terms of equipment and batteries […] and initially we have found out that this is the best acoustically[…] but “how do we mount it to the rest of the structure?”. That's a solution for the manufacturer, So they figured that out. They put some kind of metal icing here (show us the prototype), and scrape together. So that was their part, but the main design was done here.* (Adam, follow-up interview)

The purpose of the Omni source was to simplify making measurements and work efforts were invested in making the loudspeaker wireless and portable. As a natural consequence of this the company produced a suitcase specially designed to fit the loudspeaker together with a computer. All these processes have required knowledge collection and it has been necessary to gather external resources, such as manufacturing expertise, to complete the product for launch.

The activity of documentation was also essential in the work with the Omni source as the main developer had produced a paper presented at a conference where the team launched the new loudspeaker. Big parts of the work with the development of the Omni have been documented in the user manual, in excel as graphs, as well as documentation of how the development process moved forward:

> *I keep the documentation in my folder…We put all of the important files in Dropbox […] So we have access from everywhere and that's also how I organise myself generally, I use folders and subfolders for organising. […] for instance this is for the manufacturer and then we get extra things to get graphics for the manufacturer, we get different prototypes, different nodes, different measurement sessions done at different times.* (Adam, follow-up interview)

The development of the Omni source has also been characterised by a high degree of collaboration. Although it was the main developer who was described as "the anchor person" who held the project, collaboration had been central to its development and a previous student assistant was involved in designing the speaker through simulations in an external software, and helped to create prototypes:

> *I am the person who is in contact with the manufacturer, and the one who does the designs, and the…well, the designs in a way, it had started in the past when we had a student here […] we started, and we designed it together. So, both in computer and then in we did many prototypes.* (Adam, follow-up interview)

The activity of communication took place during the entire process of the development in terms of conversations and technical discussions on how so solve specific issues, but the communication activity was most prominent during the period leading up to the launch where the development of the loudspeaker was a regular feature at the weekly meetings and the main developer informed the rest of the team about the progress. Here the team gave feedback, discussed problems and shared knowledge and ideas that contributed to the further development of the project.

5.4.2 Knowledge sharing and the material calculator

In the interviews two knowledge sharing activities were particularly prevalent in developing the material calculator, that of documentation and joint knowledge collection. A large amount of work was invested by the main developer to document the development process in a single Word file, stored in a separate folder on the company's shared drive: “My intention is for it to be available if anyone wants to understand better what's going on behind” (Victor, follow-up interview). It was inspired by activities from the programming course in the engineer education and acted as a safety net: “It's more of a safety net. I would say. It's more... It's out there and people know it's out there if they need it.” (Victor, follow-up interview).

The informant had the experience that it could be hard to orient oneself in the code made by others, and it was therefore seen as easier to find back by documenting where things were and how they worked. The file did for instance explain the choices made during the trial and error-based development processes: “I tried many, many different things and then I wanted to make sure that what I did is saved somewhere” (Victor, follow-up interview). The document shared the knowledge behind the tool and was meant to be used by the author himself and others, but not just limited to the team: “it comes back a bit on how we how we handle our, how do you say like, our knowledge that we want it to be transparent so that people know exactly what's going on” (Victor, follow-up interview). While end user documentation explained what you did with the tools, development documentation explained how to make the tool.

Even though the main developer had an intention to be systematic and consistent in creating documentation few others seemed to document in the same way. This made it challenging to maintain the activity. The development process continued even after the launch of the first version of the material calculator, making the documentation process seen as never ending. The document was shared on a shared drive, but the expectations for others to engage with it were modest: “So, then it's maybe a little bit a pain to read but yeah, if somebody wants the data and then it's there” (Victor, follow-up interview).

The second activity identified from the interviews was that of collecting knowledge. As the material calculator was heavily based on acoustic theories and formulas a lot of theoretic knowledge had to be collected in its development. Beside the manager, the senior researcher was involved in the collaborative process of reading and discussing relevant literature: “[Victor] was the main person behind this tool for programming it and also together we collected background information, search literature, etc., etc.” (Jacob, follow-up interview). Promising formulas would be selected, summarised and shared in word, then tested and the test results were again shared and discussed until they were reasonably content: “Just two pages and then there's a journal reference to some of these things. Well, just a way to communicate “here's something we could look at maybe, this should be implemented”” (Jacob, follow-up interview). There were different understandings of how things would work between the experts involved in the development, which led them to propose and test different things between themselves. The process was experienced as challenging since there was no definite answer to what the best solution would be, and knowledge sharing about the object was central in developing this particular object: “the discussions and exchange of views between [Victor] and myself has been very important in this particular material calculator” (Jacob, follow-up interview). As a way of sharing the final results and part of the company culture of being open and transparent, the relevant test results and knowledge about the calculator was to be published in the scientific paper co-authored by the main developers involved. Knowledge collection were thus connected with the activities of documentation:

> *This is like all the theory and then we also backed it up with more experiments and stuff which are in the paper. So the paper is a bit, maybe more applied and this is more like, what did I try? Why did I choose this parameter over that parameter? How did I develop it and stuff like that.* (Victor, follow-up interview).

We will now analyse the challenges the expert community faced when developing the epistemic object.

5.5 Challenges

When developing the software some common challenges faced the experts in facilitating knowledge development. These can be described as dealing with the complexity and uncertainty of the objects. The complexity is related to both the complex expertise embedded in the epistemic object that the team had to struggle with, and also the complexity of how the team collaborated when developing the knowledge object. Complexity was also associated with uncertainty. Uncertainty was both a part of developing objects without having definite answers, and also experiencing uncertainty in what to develop and where the development process would lead them. This relation between the challenges of complexity and uncertainty can be demonstrated by giving three examples, the code itself, in the omni-speaker and in the material calculator.

The code itself has been building up over a period of 30 years and its complexity is vast. To orient oneself in this complexity was in itself a challenging task: “It's very, very big. And yeah, if you touched on something that you haven't touched before that already exists, like understanding what's going on, can be really tricky sometimes” (Victor, group interview). It had been maintained by the manager who over the decades had developed a deep knowledge of the code and its internal interdependencies. When changing one part of the code, another could be affected, making it a not straightforward process to integrate new code: “I think some of the integration costs of it, like how to make it fit into to do the right things all the places, it's probably a good idea that it is only one person that does it at the time” (Peter, group interview). As well as creating some uncertainty, the experience of not knowing what is where in the code and not having systematic documentation that could assist knowledge sharing could become a barrier to the development process: “It makes it quite vulnerable” (Jacob, group interview). On the other hand, the lack of documentation motivated new documentation activities. Another challenge was if the programming language had to be changed. Here the sheer complexity and size of the code would require major resources for it to be programmed into another language. So far this was an uncertainty the team had to live with, as the costs of reprogramming are deemed higher than the benefits. The uncertainty was also stabilised by the usability of the software: “what is true is that thing I think that they are working fine, and I think that's a common feeling we all have and, So if something works you just let it work” (Daniel, initial interview).

The material calculator is another example of how the team struggles with complexity and uncertainty. The calculator itself is made of several layers of materials combining different theoretical formulas in order to calculate their acoustic properties. Even though the tool itself was simplified and made user friendly, when many layers of materials based on different theories embedded were compiled in the tool, the complexity made it hard to separate the different aspects and understand how each model and their theories interacted to produce the final output. The wish to understand what is going on behind shows how the experts were trying to handle the paradox of theory not matching reality:

> *Sometimes, when you measure, you can end up with absorption coefficients that are actually more than one, which is one of the paradoxes of the method, you have something that absorbs more power than it arrives. Which is not, of course, what's happening in reality.* (Victor, follow-up interview)

The experts struggled with the uncertainty of having no definite answers in the process of developing the material calculator. Knowing what the best model was to use was a process of trial and error, comparing measurements with calculations, discussing and making choices. And even if a calculation were accurate, it was not necessarily reliable to work in any context. This uncertainty came from working with developing the complex objects, trying to make working representations of immensely complex phenomena:

> *It's really looking, really modelling the physics in there. We're not there at all (laughs), because we're trying, we're a bit more... global approach, but it is in the end the best way to go. But yeah, we cannot really have everything.* (Victor, follow-up interview).

The purpose of the development of the Omni source was to solve problems in room acoustics measurements which were common in conventional sources. The developers described in their interviews that they had experienced a variety of challenges in the process of developing the Omni:

> *…making it small is difficult, right? So you want the good sound power from the source and it should be small. And that's kind of the challenge […] and we tried different designs. A lot of different designs actually. Initially, I don't know if you've seen that one but initially made one with two speakers against each other and that*

was kind of making a very nice omni directional radiation pattern, but it didn't sound very good, so it distorted a lot. And then the problem that we saw all the time is that low frequency will cut off a lot so it's almost impossible to get any low frequency radiation. So, that's one of the key problems we had. (Peter, follow-up interview)

The developers explained how they had struggled to combine the ambition of making a small and portable speaker that also requires lots of energy and volume, and much of the uncertainty in the development of the Omni mentioned in interviews, was related to the possibility of making the source that they wanted within physical and acoustical limitations. To make it powerful and at the same time portable involved an experience of fighting the laws of physics:

So that's one thing, getting enough energy, it's really a burden. The other thing we looked at was portability […] So now you get a lot of why it looks like it does, you know one unit makes it lighter […] So, we are fighting some physics and that's always how it is with this type of source. (Peter, follow-up interview)

We will now summarise this chapter, before discussing our analysis using our theoretical framework.

5.6 Summary

Our analysis depicts a community of experts with a high level of education, where all had concluded at least a master's degree mostly in the field of acoustical engineering. The interviews indicate that the participants have a common identity from working together on developing computer simulation room acoustics software together and had complementary roles. They enjoyed a high level of independence allowing them to follow their own individual paths of learning and development. They worked on specialised tasks, but still interacted and talked together on a regular basis in order to progress. They acknowledged the value of delegation and had ways of coordinating their efforts through non-hierarchical, but efficient and interactive practices.

In our data set, two objects stood out as actual and current examples of shared epistemic objects in this community, the Omni source and material calculator. Our further analysis attended to how these objects acted as focal points for knowledge sharing, object development and epistemic practices. Knowledge sharing was central to the development of the software and its functions and activities of documentation, knowledge collection, communication and collaboration were examples of this, reflected in the development of the Omni source and material calculator. When the expert community worked on developing new functions, they faced the challenge of engaging with complex knowledge and transforming it into visual and applicable software and hardware for end users. Several external and internal drivers influenced the development of the objects. They were interconnected and came from different levels and settings moving from professional, technical, and theoretical developments in the field, to competitors and customers, before finally coming together in the internal dynamics of the team itself. The epistemic practices involved software development in iterative processes of knowledge collection and validation where functions were first developed separately before merged, integrated, and compiled into the software ready to be launched as versions that users could apply in their work. This was not straightforward, as the complexity and uncertainty of the software was a constant challenge to be handled by the expert community, both in the object itself, but also as a way of working emanating from the process of solving complex knowledge problems itself.

6. Discussion

6.1 Introduction

We will here discuss the research problem and research questions using the theoretical framework and analysis. Our broader research problem has been:

How is the development and sharing of knowledge in expert communities facilitated by working with joint epistemic objects?

The research questions contribute to answering the thesis question and are

- *In what ways does this expert community develop knowledge by way of epistemic objects and epistemic practices?*
- *How does knowledge sharing take place related to their work on epistemic objects?*
- *What challenges does this expert community face when working on joint epistemic objects?*

We will discuss findings from the analysis in relation to the theoretical perspectives and concepts presented in chapter 3, and in relation to previous research presented in chapter 2. In the following sub-chapters, we discuss each of the research questions before going back to our broader research question and discuss how our findings relate to the wider field of research.

6.2 In what ways does this expert community develop knowledge by way of epistemic objects and epistemic practices?

The epistemic objects and epistemic practices contributed to the development of knowledge in the expert community. First we discuss the team in our case as an expert community. Then in the following subchapters we discuss how knowledge was developed through joint work on epistemic objects, and how the expert community engaged in epistemic.

Expert communities engage in knowledge practices similar to those of scientific institutions in order to develop knowledge for practical appliance (Knorr Cetina & Reichmann, 2015). Members of the expert community in our case elaborated through interviews that they saw themselves as a multi-disciplinary team with an educational background in various branches of engineering, as well as physics and mathematics. Common to them all is their mutual educational background with master's or PhD from the same university as the company was located, and the teams' common interests and professional identity united them in the work of developing the room acoustic software. Merz (2018) describes how computer simulations have a mediator role by being between science and its application in the real world. The company was located at the university campus, cooperated with the university and used their facilities. In this way they interacted with the scientific community on the field and contributed to research, acting as mediators between the world of science and appliances through acoustical consulting. A prominent feature with expert communities today is that they engage in knowledge practices similar to those of scientific institutions in order to develop knowledge for practical appliance (Knorr Cetina & Reichmann, 2015). The expert community's development work was based on a strong scientific foundation from theory and research, and they engaged in knowledge practices similar to scientific institutions. They applied scientific knowledge in practice based on their professional, science-based background and individual specialities (see chapter 5.2). We will now discuss how this expert community develops knowledge by working on joint epistemic objects.

6.2.1 Expert knowledge development through epistemic objects

The computer software had many of the characteristics of an epistemic object, we will here discuss how the case relates to the concept of epistemic objects. The primary epistemic object in our case company was the room acoustics computer simulation software in the centre of the work of the expert community. The computer software had been developed for over 30 years, and it had been constantly changing and unfolding, as in the descriptions of epistemic objects by Knorr Cetina (2001). The software was created through the mobilisation of acoustical and programming expertise to answer the complex problem of how to predict acoustical phenomena in rooms not yet built as described by Forsyth (2018). Using expertise to solve complex problems is, as described by Nerland & Jensen (2012), a characteristic of object work in expert communities.

The software was constantly unfolding as it was being worked on by the expert community, acquiring new functions and improving and changing its existing ones, and in the process of developing and defining new functions there was always a future perspective (see chapter 5.2). A specific function might have been finalised for a new version, but with the expectation and understanding that it would continue to be developed. These characteristics of constantly unfolding when being worked on, is typical for epistemic objects as described by Knorr Cetina (2001). The characteristics of epistemic objects seem also to be valid with the object being a complex computer simulation software. The simulations both gave answers to existing questions of how sound behaves in rooms through its projections and generated new questions by making visible the shortcomings of the applied theories when used in concrete acoustical scenarios. Computer simulations act as a mediator between theory and experimentation, applying established theories and scientific assumptions and at the same time being open ended and experimental (Merz, 2018). The software mediated between theoretical acoustical knowledge from physics and science, and its use by acoustical engineers in open ended experimentation, generating new theoretical insight and practical knowledge published as scientific papers by the expert community (see chapter 5.3).

The software was used by different actors in different contexts fulfilling different functions, a role of computer simulations as epistemic objects according to Merz (2018). For instance, when used by clients' new questions were raised when the software did not behave as expected. The ability to generate questions are central features of epistemic objects (Knorr Cetina, 2001; Nerland & Jensen, 2012). The software was the focus of a collective exploration by the team of experts who attempted to solve the complex knowledge problems coming from client needs, developing new functions and releases, as demonstrated in weekly meetings and regular interactions between team members. The discussions of the team demonstrate how challenges with the software lead to new questions being generated through an unfolding individual and collective reflection process (see chapter 5.4).

Knorr Cetina (2001) describes epistemic objects as in a process of being defined where shortcomings of the object become visible for the experts working with the objects, which are never really completed as new lacks continuously appear as old have been solved. The software was in a process of being defined by the team of experts, as new functions need to be integrated and released with new software versions. It was however a never-ending process where scientific theories themselves had shortcomings. The complexity of the expert field of acoustics meant that the software would never be completed and work perfectly in every acoustical scenario. These shortcomings became visible to the team of experts through the testing and validation process and were a source of frustration, but also motivation to understand more what was going on behind (see chapter 5.3.2).

Knorr Cetina presents epistemic object work sometimes as a subject-object relationship (Knorr Cetina, 1997), and other times as a collective perspective (Knorr Cetina & Reichmann, 2015). In our case the expert community both worked as a team and at the same time focused on individual projects and partial objects. While the software was part of a collective development process, the experts reported delegation of tasks to individuals who worked on specific functions in the software as the main developer, even if they had strong support from their colleagues. They had a special ownership and binding to the function they developed, expressed even as that of a parent (see chapter 5.2.5), which acted as a driver for creative and constructive practice. Knorr Cetina (2001) describes how shortcomings of epistemic objects can lead to new opportunities for exploration and meaning producing. The shortcomings in the software both gave way to different opportunities of theoretical and practical exploration and drove forward a practice of meaning producing where the user and user interface perspective was central (see chapter 5.2.2).

The complexity of the software had clearly increased and new lines were continuously being added to the code when new functions were developed. Adding to the complexity the software consisted of several *partial objects*, as conceptualised by Knorr Cetina (2001). In the software, a multitude of different functions existed that functioned as partial objects. Several were coded as units, or folders in the overarching software development environment. New partial objects in the software were made as functions were improved, new functions were developed or new software releases were prepared. Even if some of the software functions could work stand-alone and independently of the software, they were a part of the software as a whole and become integrated into new versions. It was mainly the different partial objects currently being developed and discussed at meetings that drove the development forward.

Partial objects create some limits to the dynamics by coordinating the experts input and allowing for some parts to be stabilised while others are being developed (Tronsmo & Nerland, 2018). The use of partial objects may be related to the complexity of the software, where concrete functions acted as partial objects and focus points, allowing for some parts of the software to be stabilised while others were being developed, as development resources were limited and needed to be prioritised. Ewenstein & Whyte (2009) describe the function of partial objects as combining the concrete and abstract aspects of epistemic objects. The partial objects in our case were points of interaction where the expert community would work while developing different software functions, where they develop something concrete and at the same time engage with the more abstract acoustical knowledge problem.

We explored two objects presented in the analysis as partial objects, the Omni-source and material calculator. They represent two types of materiality, one being a hardware speaker to improve the measurement function of the software and the other a software function used to calculate material properties. While both represented acoustical knowledge problems, the development triggered challenges specific for each requiring specific types of expertise and knowledge to be developed. As epistemic objects they both represent complex knowledge problems and acted as focal points for unfolding processes of exploration, leading to new incomplete partial epistemic objects that signify new potential directions and opportunities for development. While in the speaker the acoustical problem revolved around finding the best physical attributes to create an omni-directional, radiating sound signal using only one speaker, the material calculator tried to identify the best theoretical basis for calculating complex material properties of sound absorption and reflection.

The development process involved manipulating and exploring the properties of the epistemic object through these partial objects. When developing the omni-source it was experimenting with different constructions, shapes and hardware settings, when developing the material calculator, it was testing different models and comparing with measurements. Knorr Cetina (1997) describes how partial objects have a signifying force directing future exploration of the epistemic object. The signifying force of the partial objects of the software directed future exploration and ways of extending practice by showing different paths the experts could follow, like when a category of theories for the material calculator gave promising test results, or when the initial idea of having two speaker elements in the source didn't work and a solution with one speaker, and the idea of a lens that could spread the sound in an omni-directional pattern came up. According to Werle & Seidl (2015) partial objects represent something larger than the object itself and at the same time acts as a concrete object where experts can interact and engage in exploration that generates interest and motivation. The partial objects represented the larger field of acoustical expertise and were at the same time concrete objects that the team could explore and generated great interest and motivation for in the development of new knowledge.

Materialising an epistemic object stops it from continuously changing by ignoring the lacks and discrepancies of the object so it gains a material identity making it possible to use for a particular purpose (Knorr Cetina, 2001). The software and its functions had to be materialised in temporary or final versions in order to be used by acoustical consultants and become starting points for further development. With the material calculator two versions were made, one development version and one final version for the latest software release. In the final version unfinished properties were hidden or removed so users could use the calculator with the most reliable results. With the Omni source prototypes were used to work out final lacks to achieve an omni directional radiation pattern, so it could be used in professional measurements. Knorr Cetina (2001) describes how software as an epistemic object is materialised in familiar shapes such as versions and updates and how developers put in special efforts to meet users' needs. With the software as a whole, the modified code and functions developed in different partial test versions and environments were finalised and integrated into new versions of the software. The software then acted as a materialisation in a format familiar to the users and that met the needs of the user community the software served, as described by Knorr Cetina (2001). The software no longer worked as an epistemic and question generating object, but could be used as a tool by acoustical consultants as a technical object, reflecting the epistemic objects ability to shift from epistemic to technical objects when materialised as described by Ewenstein & Whyte (2009, p. 12).

The software development process switched between the process of exploration, where developing and improving functions lead to knowledge development, and confirming processes where software was materialised and new releases and patches were issued. This is in line with Nerland & Jensen (2012) description of how epistemic objects can engage experts in changing processes of exploration and confirmation. In the software development team these processes happened in parallel, as the diversity of tasks involved both routine support and maintenance as well as projects of innovative development. Development of new versions and monitoring of existing ones were a *continuous* process such as the software processes described by Fitzgerald & Stol (2017) where deployment and monitoring processes take place in frequent intervals. In the analysis we found that monitoring releases and receiving feedback triggered new developments and new partial objects. We will now look at how these epistemic objects were developed through the expert community engaging in epistemic practices.

6.2.2 Expert knowledge development through epistemic practices

When developing the software, the expert communities engaged in epistemic practices. We will here summarise the findings related to drivers and activities of knowledge sharing from our analysis, in relation to theory on epistemic practices of engineers as reflected in Cunningham & Kelly (2017). We will also discuss the theoretical implications of what we have found.

By working on developing the software the expert community proposed, communicated, assessed and legitimised knowledge claims in socially organised and interactionally accomplished ways, as described in the definition of epistemic practices by Cunningham & Kelly (2017). According to Nerland & Jensen (2012) epistemic objects structure the practices of exploring complex knowledge problems, accessing knowledge resources, identifying open opportunities and getting direction for further explorations, practices recognisable in the development of the software in the case company. As expert communities they took part in *wider machineries of knowledge construction* that explain how responsibilities are distributed in extended networks where practitioners, researchers and agencies interact in developing and circulating expert knowledge on a macro, meso and micro-level, as described by Nerland & Jensen (2012).

As acoustical engineers they participated in epistemic practices of engineers and several epistemic practices were identified in the analysis. They acted as drivers in the development of epistemic objects (see chapter 5.2) and were supported by knowledge sharing activities (see chapter 5.4). They have been placed where they could best inform the categories of epistemic practices of engineers, as seen in Table 6. The expert community used knowledge sharing actively in the practice of developing software and according to Liu (2019), epistemic objects coordinate the fragmented and heterogenous knowledge in expert communities.

	Category of epistemic practices of engineers	Drivers of knowledge development and *Knowledge sharing activities*
1	Engineering in social contexts - Parameters set by clients and conditions. - Inner workings of engineering teams	External drivers Internal drivers *Documentation*
2	Uses of data and evidence to make decisions	External drivers *Knowledge collection*
3	Tools and strategies for problem solving	Internal drivers *Collaboration*
4	Finding solutions through creativity and innovation	*Communication*

Table 6: Epistemic practices of engineering

The different practices in the table also involve the different levels of the wider machineries of knowledge construction, but as we can draw from our analysis these levels are interconnected and come together in the practices of developing partial objects, as with the Omni source and material calculator. This is in line with Knorr Cetina (Knorr Cetina, 1999, 2001) understanding of how different elements are joined together in particular ways in local practices. In epistemic practices the expert community both show agency by applying generalised expert knowledge to specific cases (meso level), specific actors verify expert knowledge and take on specific tasks in larger knowledge contexts (macro level) and epistemic objects assist processes of knowing and have a transformative and stabilising role (micro level) as described by Nerland & Jensen (2012). While the levels are joined together in epistemic practice, we see for instance that analytically, category one could highlight the meso level, category two the macro level and category three the micro level.

Engineering in social contexts is a category of epistemic practices of engineers and refers to how the problem and knowledge that the software is a response to originate from social needs and are solved through social processes including evaluation by clients as described by Cunningham & Kelly (2017), in this case the users of the software. The social need is the need to have reliable prediction of sound acoustics in rooms. The client evaluation is the feedback from customers and other users and stakeholders of the software. We would here include users and stakeholders of the software, because according to Cunningham & Kelly

(2017) a unique feature of engineering that makes it different from other disciplinary knowledge is that knowledge is developed through practices where requirements are defined by clients and the relevant conditions of the situation where the design work takes place. The category has two subcategories. *Parameters set by clients and conditions* are external practices. In the analysis external drivers were stakeholders that used the software and had a channel where they could give input, and the problems and solutions identified here were the ones located in specific contexts external to the team (see chapter 5.2.2). *Inner workings of engineering teams* deal with how engineering took place socially in the team. In the analysis internal drivers were the inner discussions and dynamic processes of the team leading to the decision of what to develop next (see chapter 5.2.2). We also place the knowledge sharing activity of documentation here, as it supported the coordination of internal and external drivers (see chapter 5.4)

Uses of data and evidence to make decisions is the second category of epistemic practice in engineering and involves the importance of data and evidence when engineers construct knowledge to solve problems, including understanding user needs, assessing problems, testing parameters, building prototypes and presenting solutions to clients as described by Cunningham & Kelly (2017). From our analysis this involves both the external drivers on a macro level (see chapter 5.2.2) and the knowledge sharing activities of documentation and knowledge collection (see chapter 5.4). Cunningham & Kelly (2017) describe how engineers collect data and evidence to inform design when theory is not reliable to accurately predict the performance of the design. The expert community lacked theory that would work perfectly in any circumstance engaged in data-collection to inform design of the software to get more reliable predictions from simulations (see chapter 5.3.1 and 5.3.2).

Tools and strategies for problem solving is the third category of epistemic practice in engineering and refer to the evolving tools and strategies used in problem solving, providing analysis and designing solutions and embodying knowledge in the solutions as described by Cunningham & Kelly (2017). From our analysis we place here the knowledge sharing activity of collaboration, involving problem solving with colleagues, material considerations (see chapter 5.4), systems thinking (see chapter 5.2.3) and building prototypes (see chapter 5.3.1).

The fourth category of epistemic practice in engineering is *finding solutions through creativity and innovation*, stressing the importance of creativity for innovation and finding solutions in the highly context dependent reasoning processes in engineering. From our analysis we place the knowledge sharing activity of communication here (see chapter 5.4). This is the least elaborated engineering epistemic practice category by Cunningham & Kelly (2017) and a category where many of the other practices connect.

One explanation of how this takes place may be related to the highly context dependent reasoning process in engineering, meaning that the practice of developing processes has a character that is unique for the specific group of engineering experts and the conditions of the problem. We saw two different examples of this in the analysis, where the material calculator was developed in a different way than the omni-source (see chapter 5.3.2). The processes were the result of conscious practices evolving depending on both what was being developed and by who it was developed, as they had different specialities. The omni source was developed first based on the managers' ideas, tested through a student assistants project and the developing experts' mechanical background and interest in design. This can be described as a rather unpredictable path to solve the challenge of designing the speaker unit (see chapter 5.3.1). The material calculator was built on the PhD of the main designer, and a long history of research on material properties by the senior researcher (see chapter 5.2.5). Problems were solved as they appeared and the processes evolved in an unpredictable way, where processes needed to solve the problem at hand were developed as they went (see 5.3.2). The epistemic object structured and coordinated the unfolding process of problem solving as described by Knorr Cetina (2001), bringing together the different knowledge of the team of acoustical engineers, and the wider field of expertise as described by Nerland & Jensen (2012) in epistemic practices.

6.3 How does knowledge sharing take place related to their work on epistemic objects?

The expert community in our case developed specific ways to create and share knowledge both with the group and the wider community by exploring, developing, and mobilising different epistemic objects. This resembles the practices of expert communities, as depicted by Knorr Cetina & Reichmann (2015) and Nerland & Hasu (2021). On a group level, several epistemic practices of knowledge generation and sharing manifested themselves through the team's ways of working. These practices were not dictated from a management level but occurred organically from having worked together for many years. The practices were supported by the knowledge sharing activity of communication which acted as a key factor to keep each other updated and give professional feedback on projects, and was specifically mentioned in most of the interviews as an essential part of knowledge sharing (see chapter 5.4).

Epistemic objects facilitate knowledge sharing by coordinating fragmented and heterogeneous knowledge and have a productive and enabling role in triggering the development of joint transactive memory in expert communities (understood as knowing who knows what in the team) and increasing collective sensemaking (Liu, 2019). The weekly meetings were an example of how communication between team members flowed in coordinated ways, mixing the fragmented knowledge of the group both through task distribution as well as problem solving processes. While most of the team shared educational background, each had their own specialties (see chapter 5.2). Transactive memory can be achieved through frequent interactions between team members (Liu, 2019). The teams' development of transactive memory manifested itself through their focus on active communication which was an essential part of their daily routine and became visible through their planned meetings, as well as several spontaneous knowledge sharing activities that took place when needed (see chapter 5.4). According to Wang & Noe (2010) high cohesiveness and long formed teams support knowledge sharing. In our study, the team had been formed for a long time and experienced high cohesiveness. The low threshold for giving feedback and accessibility of the team, supported by searching for knowledge online, seemingly speeded up innovation processes (see chapter 5.4).

Team communication as a planned knowledge sharing activity is best described in weekly meetings, which were of an informal character and the word was open to everyone. From our understanding this provided a safe environment for sharing thoughts and ideas with the others. The team expressed that they benefited from having the meeting as it provided a dynamic conversation, creating and sharing knowledge by discussing issues that could lead to new improvements or developments of the software. The experts also highlighted the benefits of knowing what’s going on with the others and what they were working on in case they had faced a similar issue themselves or knew how to fix a problem that a college was struggling with. The open communication provided an access to each other’s work and competence that we would argue positively contributes to their knowledge sharing and development practices (see chapter 5.4).

Knowledge was also shared and developed through the creation of documentation that represented knowledge about the object, such as with the material calculator and omni source (see chapter 5.4). Documenting the development of the software made the knowledge available to the others on the team, enabling them to follow the progress and have insight in order to contribute to the development of objects. Although such documentation practices facilitated knowledge sharing, there were only parts of the expert community who were fully invested in this practice and saw how they as a team could benefit from documenting their work. It's possible to argue that a more holistic approach to documentation practices could contribute to increasing the teams' knowledge on each other's processes and facilitate a transparent practice regarding knowledge sharing and development.

Another prominent activity that facilitated knowledge sharing and development was collaboration, which manifested itself through their daily interactions such as turning to a colleague to discuss a problem (see chapter 5.4). In our view, this was a key factor in their knowledge sharing and development practices, and we would argue that the active communication and their shared understanding of depending on each other positively contributed to their work. The shared understanding that they could not know and understand everything individually was reflected in the practices of always having two people involved in development projects and appeared in line with their practice of sharing competence through daily interactions that led to the creation of innovative ideas for development of the software.

As Knorr Cetina & Reichmann (2015) explains, expert communities form around the knowledge problems linked to specific areas of expertise, and through interviews with the members of this expert community, acoustical engineering was identified as the core disciplinary identity within the team, even by the non-acousticians. Their shared identity and belonging to the field of expertise is reflected through the teams' contributions to the wider machineries of knowledge construction on a meso level (Nerland & Jensen, 2012), by their participation at conferences where they represent the company and present contributions in forms of research papers and presentations of new developments, such as the omni source that was recently launched.

Alongside the above mentioned practices, the expert community also facilitated workshops and courses developed for their end users, and in this way took part in what Knorr Cetina & Reichmann (2015) described as processes of learning and knowledge creation linked to wider knowledge cultures specific for their expert areas. The company provides courses adapted to the user's knowledge levels of the software, by facilitating full weeks dedicated to software training combining demonstrations, hands-on exercises and discussions on specific cases brought by the participants, which contributes to the expert community's practices related to the creation, development and sharing of knowledge as described by Nerland & Hasu (2021).

6.4 What challenges does this expert community face when working on joint epistemic objects?

In our analysis we identified the two challenges of complexity and uncertainty, and exemplified them with reference to the code, the Omni source and the material calculator.

As described by Knorr Cetina (2001) complexity is a characteristic of epistemic objects, their complexity increases as it is being worked on and knowledge of the object increases and they introduce more complexity and variation to the area of expertise. In our findings the expert community experienced complexity from the increasing size of the code and manifold of functions gathered in the software, requiring the ability to focus on partial objects and find ways of handling the complexity. They also experience complexity in the difference between the theory implemented in the software and the results of measurements in actual rooms, requiring repeated testing and acceptance of uncertainties. The increasing complexity they experience supports our claim that the software is an epistemic object and that knowledge is being developed. The complexity also opens up for a range of opportunities for the object to be explored, and in this sense complexity can act as a driver for new developments and can end up in new functions. Nerland & Jensen (2012) describe this as *explorative* practices where the object unfolds and complementing *confirmative* practices where the objects are materialised in order to be used, such as when our expert community were finalising partial objects for launch.

As we found in our analysis, complexity and uncertainty were interconnected and uncertainty became an issue when the expert community was faced with the great complexity of the field of expertise. Uncertainty was connected to developing objects without definite answers, as well as uncertainty in what to develop and where the process would lead them (see chapter 5.2.2). Dingsøyr et al. (2012) describes uncertainty as an inherent part of agile software development that is futile to attempt to control. Uncertainties also occur in room acoustical design practice and are related to the uncertainties in estimations and algorithms (Brinkmann et al., 2019) and is a central characteristic of computer simulations in room acoustics (Vorländer, 2013). Uncertainty in our case might thus be embedded in the field of expertise itself, and as mentioned by one of the experts on the team, never being completely sure was the biggest challenge (see chapter 5.3.2). Uncertainty can also take the form of *task uncertainty*, where lacking necessary information to develop the software hinder effective knowledge sharing (Ghobadi, 2015). Task uncertainty might as well be an inevitable part of epistemic practices, as a developer in our expert community mentioned, their task was to come up with the tasks (see chapter 5.2.1). This was central when developing knowledge such as coming up with and developing new functions. Identifying tasks took place both during discussions in the weekly meetings and through collaboration between team members, where the manager was a central actor. Uncertainty seems to be connected with the lacks and undefinable character of epistemic objects, but can be handled like in Ewenstein & Whytes' (2009) article where uncertainty helped focus team efforts in finding solutions through subtle processes.

Nerland & Jensen (2014) saw the need for more research on the ways expert communities deal with complexity and uncertainty. An example of how this can take place is given by Mahringer et al. (2019) who argue that routine helped restrict and coordinate unfolding processes of epistemic objects through practices of structuring, materialising, recalling and closing. They found that balancing the need of order and wish to explore created tensions and negative emotional reactions, requiring positive emotions to achieve emotional balance. Dealing with complexity and uncertainty was also central in practice of our case company. It was challenging experiences of fighting physics where results didn't always make sense (see chapter 5.5), but how they describe these challenges with enthusiasm gives the impression that the expert community have reconciled with the complexity and uncertainty as an integrated part of working with epistemic objects, and succeeding in developing stable functions was a motivational factor (see chapter 5.2.5). The omni-source needed to be produced and sold and the material calculator had to be implemented into the software and be usable for end users.

The main strategy for dealing with complexity and uncertainty in our case can be drawn from the ways of working of the team. Communicating and talking together were commonplace and contributed to confidence on the team (see chapter 5.4) and involved discussing and agreeing on the direction. Sharing the responsibility for decisions made acted as a social mechanism in dealing with complexity and uncertainty by helping support the unfolding process and was dependent on frequent communication and interactions. It reflects the epistemic practice of engineers of *envisioning, constructing and assessing multiple solutions* where solutions gradually emerge through dialogic conversations considering different responses, as described by Cunningham & Kelly (2017). While the epistemic object is in the centre, its complexity elicits uncertainty in the experts when worked with. Strategies such as social leverage were supported by transactive memory (Liu, 2019) and helped counter the uncertainties and stop them from becoming barriers of knowledge sharing, such as with the task uncertainty described by Ghobadi (2015). Then the complexity of epistemic objects can rather open possibilities for developing new knowledge as described by Nerland & Jensen (2012).

6.5 How is the development and sharing of knowledge in expert communities facilitated by working with joint epistemic objects?

We have identified the specific ways knowledge sharing and development take place by exploring the role of epistemic objects and epistemic practices in a specific expert community developing a software for room acoustics computer simulation. We will here discuss how our findings relate to the wider field of research.

Joint work on epistemic objects facilitated epistemic practices in software development. Dittrich et al. (2020) showed how software development doesn't necessarily need to follow a specific methodology, but evolve as the results of deliberate practices. We have here given an example on software development as epistemic practices in room acoustics simulation and showed how rather than the software development methodologies themselves, acoustical expertise and the background of the experts shaped the practices (see chapter 5.2) as *evolving practices* over time, as described by Dittrich et al. (2020).

The expert community had formed over a long time and showed how older development teams work in practice and have been discussed using an established theoretical framework, as called for by Dingsøyr et al. (2012). We used the concepts of epistemic objects and practices as described by Knorr Cetina and others (see chapter 3) and analysed a case of software development practices similar to those of Fitzgerald & Stol (2017) label continuous. Fitzgerald & Stol (2017) asked how continuous evolution and maintenance of software systems can be facilitated and how key barriers between development and operations can be removed. We here gave an example of how the software evolved for over three decades facilitated by changing epistemic practices used in working with software as an epistemic object, and how working in a small team reduced barriers between development and operations (see chapter 5.3).

We identified four knowledge sharing activities that contributed positively to the expert community's epistemic practices used in developing the software as an epistemic object. Wang & Noe (2010) asked for more research on how expert communities establish knowledge sharing cultures and how these affect dynamics of knowledge development. We have analysed how software as an epistemic object supported knowledge sharing activities such as collaboration and communication in a highly cohesive team environment. Further, Ghobadi (2015) argues that understanding the drivers of knowledge sharing in software development teams helps members reach ideal levels of knowledge sharing and communication. We have identified internal and external drivers of epistemic objects that support knowledge sharing in a team of experts developing highly specialised software. Technology-related and task related knowledge sharing drivers in software development were the least developed of Ghobadi's ((2015) categories. The concept of epistemic objects can inform the technology-related category, and the uncertainty experienced by the expert community can inform the task-related category, where transactive memory, collaboration and knowledge collection activities support overcoming the lack of necessary information in task-uncertainty. Cunningham & Kelly (2017) saw the need for studying knowledge generating practices based on the social meaning making processes to understand how engineers develop knowledge. We have contributed by giving an example of engineering practices in highly specialised software development where the end users play a central role.

Software as an epistemic object elicits specific epistemic practices and ways of working. According to Merz (2018), computer simulations act as mediators between theory and experimentation. In computer simulation of room acoustics this is clearly the case, exemplified with the material calculator mediating between scientific models and experimentations in simulations representing different real-life scenarios. The challenges in room acoustics computer simulation software listed by Vorländer (2020) involved uncertainties embedded in the project of trying to make reliable and realistic simulations. We have analysed the complexity of solving problems of room acoustics computer simulations and how expert communities tackle these challenges and uncertainties through epistemic practices.

Finally, joint epistemic object work in our expert community involved epistemic practices of engineering where knowledge sharing was an integrated part. According to Nerland & Jensen (2012), the concept of epistemic practices can be advanced by seeing how these are carried out in different expert communities. We have identified epistemic practices in an acoustical expert community and discussed these by using the framework of epistemic practices in engineering developed by Cunningham & Kelly (2017) and discussed how wider machineries of knowledge construction come together in epistemic practices, as described by Nerland & Jensen (2012). We have here shown how knowledge sharing is part of epistemic practices and facilitated by the epistemic objects ability to elicit knowledge sharing by coordinating knowledge development (Liu, 2019). This may add to a heuristic of strategies used in epistemic practices, as called for by Nerland & Jensen (2012).

Nerland & Jensen (2012) asks for more research on how practitioners negotiate what problem to focus on and what investigation-strategy to choose when several paths are possible. We found examples of expert communities making priorities of what functions to develop (se chapter 5.2), involving knowledge sharing activities of communication and collaboration (see chapter 5.4), where agreeing on what direction to take is the result of complex decision making processes, and as described by Cunningham & Kelly (2017) involving considering possibilities and restrictions related to clients, technology and theory, as well as making trade-offs between criteria and constraints by balancing needs and realistic constraints in design work. In the next chapter we will summarise our findings and discuss their significance.

7. Conclusion

7.1 Summary

We have explored how knowledge sharing and development takes place in highly specialised knowledge work by asking the broader question: *how is the development and sharing of knowledge in expert communities facilitated by working with joint epistemic objects?* Through observing and interviewing a team of eight experts in a small company that collaborates on developing a room acoustic simulation software, we found that they function as an expert community whose work was embedded in a wider knowledge culture within their field, and they participated in specific knowledge cultures to develop the software. The specialists engaged with the software as an epistemic object to solve complex acoustical knowledge problems in unfolding processes of exploration and stabilisation, applying scientific knowledge to concrete cases. Working with epistemic objects elicited knowledge sharing that coordinated the knowledge of the expert community. They worked on partial epistemic objects, such as the omni-source and material calculator, each functioning as focus points for developing the software and contributing to the wider field of expertise. By engaging in epistemic practices of engineering they shared knowledge and solved complex knowledge problems linked to wider machineries of knowledge construction in practices of *engineering in social contexts, uses of data and evidence to make decisions, tools and strategies for problem solving and finding solutions through creativity and innovation*. They faced challenges of complexity and uncertainty originating from working on highly complex epistemic objects and employed social strategies of communication and collaboration in the highly cohesive team environment to handle these challenges. We will now explore the implications of our findings, identify further research possibilities and declare the limitations of our study.

7.2 Implications

Working with epistemic objects is a characteristic of expert communities and engaging in such epistemic practices can be challenging. Expert communities of engineers such as the one explored in this thesis engaged in problem solving from a practical perspective. While engaging in these practices were implicit to the expert communities, explicitly looking at own practices can help expert communities to increase awareness about how they work and evaluate their own practices. Comparing own practices to such frameworks as that of Cunningham & Kelly (2017) may support this development and see how and where practices can be improved. Challenges in our case were connected with complexity and uncertainty, but from theory we could see that these were innate characteristics of working with epistemic objects that both should and could not be avoided. Instead, expert communities need to facilitate strategies to cope with challenges and live with the complexity. As in our case, being part of a highly cohesive community can support coping with complexity and uncertainty, relieving the experts from the pressure coming from working with epistemic objects. Knowledge sharing activities such as communication, collaboration, knowledge collection and documentation are central to coping with complexity and uncertainty. Reassurance from having discussed with a colleague, agreed on a direction, engaged with the wider expert field and documented the development can help expert communities navigate complexity and reach a point of materialisation where the object can be applied by others in practice. Companies and expert communities can facilitate this by ensuring regular interactions and close location of colleagues, such as weekly meetings and collocated offices, and encourage flat hierarchy and common problem solving in daily work, as well as take part in the wider machineries of knowledge construction by attending conferences and collaborating with universities.

7.3 Future research

While we have researched one epistemic object in the field of acoustical engineering and competing software providers are likely to have their own specific practices of software development, linking into the wider machineries of knowledge construction of our case. Comparative studies could contribute to a more comprehensive understanding of the specialised field by exploring how competing software interpret the epistemic object differently and engage in similar or different epistemic practices (Nerland & Jensen, 2012). The different levels of the machineries of knowledge constructions as described by Nerland & Jensen (2012) could also be further explored, such as the meso level, to achieve a more holistic view on the relationships between different actors involved in epistemic work.

Further, we have studied a team with a long history of software development, but only over a limited period of time. Longitudinal studies of expert communities such as ours can give a deeper understanding of how software development practices change over time, as described by Dittrich et al. (2020). Here, more use of observation and following the team in the different activities they engage in can give reliable data that are less influenced by the interpretations given in interviews.

Finally, while this study we looked at the theoretical implication of a framework of epistemic practices in engineering (Cunningham & Kelly, 2017), other professions have other epistemic practices and theory on epistemic practices could benefit from the development of such frameworks for other expert communities and comparing these frameworks with that used in this thesis. The theoretic framework should also be applied to other software engineering expert communities and other professions that work with epistemic objects, in order to broaden the understanding of how epistemic objects facilitate knowledge sharing and knowledge development, and if these practices have commonalities and differences with our case.

7.4 Limitations

The findings of this study have to be seen in light of some limitations, where the first limitation applies to the time aspect as the majority of this study has been planned, carried out and written over a period of six months. This has caused some limitations regarding our empirical work. Due to time limitations, we only had empirical access to the team for a period of six weeks, which is a short amount of time considering that our case company's history of development reaches 30 years. A natural consequence of our limited data collection period is that our empirical foundation can only describe the teams’ practices during the time we had access to them, and didn’t provide us with the opportunity to follow the development of the software and the teams’ activities over time. The second limitation concerns the observation material. During our data collection period we mainly observed one specific activity for knowledge sharing and development, being the team's weekly meeting. As a consequence, we gained limited insight into their everyday work besides this particular meeting activity. Furthermore, a limitation can be seen in connection with digital data collection. A majority of our interviews and observations took place digitally via zoom, and it is possible that this has had an impact on the communication and dynamics of the interview, which can affect the findings.

The final limitation to our findings is that our selection of participants is composed of a specific, non-generalisable group of experts with a high level of competence within their scientific field. Due to a highly specialised case studied in a specific environment, these findings are not valid beyond this study and is not transferable to other software development teams or teams of acoustical engineers.

8. Literature

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Brinkmann, F., Aspöck, L., Ackermann, D., Lepa, S., Vorländer, M., & Weinzierl, S. (2019). A round robin on room acoustical simulation and auralization. *The Journal of the Acoustical Society of America*, *145*(4), 2746–2760. https://doi.org/10.1121/1.5096178

Bryman, A. (2016). *Social research methods* (5th ed.). Oxford University Press.

Creswell, J. W. (2013). *Qualitative inquiry and research design: Choosing among five approaches* (3rd ed). SAGE Publications.

Cunningham, C. M., & Kelly, G. J. (2017). Epistemic Practices of Engineering for Education. *Science Education*, *101*(3), 486–505. https://doi.org/10.1002/sce.21271

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software, 85*(6), 1213–1221. https://doi.org/10.1016/j.jss.2012.02.033

Dittrich, Y., Michelsen, C. B., Tell, P., Lous, P., & Ebdrup, A. (2020). Exploring the evolution of software practices. *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 493–504. https://doi.org/10.1145/3368089.3409766

Ewenstein, B., & Whyte, J. (2009). Knowledge Practices in Design: The Role of Visual Representations as `Epistemic Objects'. *Organization Studies*, *30*(1), 07–30. https://doi.org/10.1177/0170840608083014

Fitzgerald, B., & Stol, K.-J. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software, 123*, 176–189. https://doi.org/10.1016/j.jss.2015.06.063

Forsyth, C. (2018). A Method for Virtual Acoustic Auralisation in VR. *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*, 1–3. https://doi.org/10.1145/3243274.3243304

Ghobadi, S. (2015). What drives knowledge sharing in software development teams: A literature review and classification framework. *Information & Management*, *52*(1), 82–97. https://doi.org/10.1016/j.im.2014.10.008

Knorr Cetina, K. (1997). Sociality with Objects: Social Relations in Postsocial Knowledge Societies. *Theory, Culture & Society*, *14*(4), 1–30. https://doi.org/10.1177/026327697014004001

Knorr Cetina, K. (1999). *Epistemic Cultures: How the Sciences Make Knowledge*. Harvard University Press. https://doi.org/10.2307/j.ctvxw3q7f

Knorr Cetina, K. (2001). Objectual practice. In *The Practice turn in contemporary theory*. Routledge.

Knorr Cetina, K., & Reichmann, W. (2015). Epistemic and learning cultures: Wohin sich Universitäten entwickeln. In *Epistemic and learning cultures: Wohin sich Universitäten entwickeln* (pp. 44–61). Beltz Juventa.

Kumar, G., & Bhatia, P. K. (2012). *Impact of Agile Methodology on Software Development Process*. *2*(4), 6.

Liu, J. (2019). Trust trigger and knowledge elicitor: The role of epistemic objects in coordinating the fragmentation and heterogeneity of knowledge in digital innovation networks. *Knowledge and Process Management*, *26*(4), 332–345. https://doi.org/10.1002/kpm.1613

Mahringer, C. A., Dittrich, K., & Renzl, B. E. (2019). Interdependent Routines and Innovation Processes – An Ethnographic Study of Scrum Teams. *Academy of*

Management Proceedings, *2019*(1), 11891. https://doi.org/10.5465/AMBPP.2019.11891abstract

Manifesto for Agile Software Development. (n.d.). Retrieved 28 June 2022, from https://agilemanifesto.org/

Merz, M. (2018). Epistemic Innovation. In W. Rammert, A. Windeler, H. Knoblauch, & M. Hutter (Eds.), *Innovation Society Today* (pp. 325–339). Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-19269-3_15

Nerland, M., & Hasu, M. (2021). Challenging the belief in simple solutions: The need for epistemic practices in professional work. *Medical Education*, *55*(1), 65–71. https://doi.org/10.1111/medu.14294

Nerland, M., & Jensen, K. (2012). Epistemic practices and object relations in professional work. *Journal of Education and Work*, *25*(1), 101–120. https://doi.org/10.1080/13639080.2012.644909

Nerland, M., & Jensen, K. (2014). Changing Cultures of Knowledge and Professional Learning. In S. Billett, C. Harteis, & H. Gruber (Eds.), *International Handbook of Research in Professional and Practice-based Learning* (pp. 611–640). Springer Netherlands. https://doi.org/10.1007/978-94-017-8902-8_23

NVivo. (n.d.). Retrieved 7 May 2022, from https://www.qsrinternational.com/nvivo-qualitative-data-analysis-software/home

Silverman, D. (2017). *Doing qualitative research* (5th ed.). SAGE Publications Ltd.

Thagaard, T. (2018). *Systematikk og innlevelse en innføring i kvalitative metoder*. Fagbokforlaget.

Thompson, J. (2022). A Guide to Abductive Thematic Analysis. *The Qualitative Report*. https://doi.org/10.46743/2160-3715/2022.5340

Tjora, A. H. (2017). *Kvalitative forskningsmetoder i praksis* (3rd ed.). Gyldendal Akademisk.

Tronsmo, E., & Nerland, M. (2018). Local curriculum development as object construction: A sociomaterial analysis. *Teaching and Teacher Education*, *72*, 33–43. https://doi.org/10.1016/j.tate.2018.02.008

Vorländer, M. (2011). *Models and algorithms for computer simulations in room acoustics*. International Seminar on Virtual Acoustics, Valencia. http://sea-acustica.es/fileadmin/publicaciones/ISVA_005.pdf

Vorländer, M. (2013). Computer simulations in room acoustics: Concepts and uncertainties. *The Journal of the Acoustical Society of America*, *133*(3), 1203–1213. https://doi.org/10.1121/1.4788978

Vorländer, M. (2020). *Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality*. Springer International Publishing. https://doi.org/10.1007/978-3-030-51202-6

Wang, S., & Noe, R. A. (2010). Knowledge sharing: A review and directions for future research. *Human Resource Management Review*, *20*(2), 115–131. https://doi.org/10.1016/j.hrmr.2009.10.001

Werle, F., & Seidl, D. (2015). The Layered Materiality of Strategizing: Epistemic Objects and the Interplay between Material Artefacts in the Exploration of Strategic Topics: The Layered Materiality of Strategizing. *British Journal of Management*, *26*, 67–89. https://doi.org/10.1111/1467-8551.12080

Appendices

A. Privacy Representative Approval from Norwegian Centre for Research Data

6/17/22, 12:33 PM Meldeskjema for behandling av personopplysninger

Notification form / Ekspertfellesskap i praksis - kunnskapsutvikling gjennomepistem... / Assessment

Assessment

Reference number
537110

Project title
Ekspertfellesskap i praksis - kunnskapsutvikling gjennomepistemiske objekter

Data controller (institution responsible for the project)
Universitetet i Oslo / Det utdanningsvitenskapelige fakultet / Institutt for pedagogikk

Project leader
Monika Bærøe Nerland

Student
Maja Greni Sundland

Project period
01.12.2021 - 31.12.2022

Notification Form

Date	**Type**
30.11.2021	Standard

Comment
Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet med vedlegg den 30.11.2021. Behandlingen kan starte.

TYPE OPPLYSNINGER OG VARIGHET
Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 31.12.2022.

LOVLIG GRUNNLAG
Prosjektet vil innhente samtykke fra de registrerte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse som kan dokumenteres, og som den registrerte kan trekke tilbake. Lovlig grunnlag for behandlingen vil dermed være den registrertes samtykke, jf. personvernforordningen art. 6 nr. 1 bokstav a.

PERSONVERNPRINSIPPER
NSD vurderer at den planlagte behandlingen av personopplysninger vil følge prinsippene i personvernforordningen om:

- lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til behandlingen
- formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke viderebehandles til nye uforenlige formål
- dataminimering (art. 5.1 c), ved at det kun behandles opplysninger som er adekvate, relevante og nødvendige for formålet med prosjektet
- lagringsbegrensning (art. 5.1 e), ved at personopplysningene ikke lagres lengre enn nødvendig for å oppfylle formålet

DE REGISTRERTES RETTIGHETER
NSD vurderer at informasjonen om behandlingen som de registrerte vil motta oppfyller lovens krav til form og innhold, jf. art. 12.1 og art. 13.

Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18) og dataportabilitet (art. 20).

Vi minner om at hvis en registrert tar kontakt om sine rettigheter, har behandlingsansvarlig institusjon plikt til å svare innen en måned.

FØLG DIN INSTITUSJONS RETNINGSLINJER
NSD legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1. f) og sikkerhet (art. 32).

For å forsikre dere om at kravene oppfylles, må dere følge interne retningslinjer og eventuelt rådføre dere med behandlingsansvarlig

https://meldeskjema.nsd.no/vurdering/61925a1c-3705-4062-a2e3-7732e9ccd869 1/2

institusjon.
MELD VESENTLIGE ENDRINGER
Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til NSD ved å oppdatere meldeskjemaet. Før du melder inn en endring, oppfordrer vi deg til å lese om hvilke type endringer det er nødvendig å melde: https://www.nsd.no/personverntjenester/fylle-ut-meldeskjema-for-personopplysninger/melde-endringer-i-meldeskjema Du må vente på svar fra NSD før endringen gjennomføres.

OPPFØLGING AV PROSJEKTET
NSD vil følge opp ved planlagt avslutning for å avklare om behandlingen av personopplysningene er avsluttet.

Kontaktperson hos NSD: Henning Levold
Lykke til med prosjektet!

B. Info letter

Would you like to participate in the research project *"Knowledge sharing in expert communities – exploring collaborative work with joint epistemic objects"*

This is an inquiry about your participation in a research project where the purpose is to explore how knowledge sharing and knowledge development and collaboration in teams takes place in expert communities. In this letter we provide you with information about the objectives of the projects and what participation will entail for you.

Purpose
The purpose of the project is to explore how knowledge development and knowledge sharing, and collaboration takes place in highly specialised knowledge work, based on the case of your company and software. The research question is: *How is the development and sharing of knowledge in expert communities facilitated by working with joint epistemic objects?*

Through interviews and observations over an approximately six-week period we are interested in exploring how your software contributes to knowledge sharing and what challenges the participants face when collaborating to develop the software as an epistemic object.

The project is a master thesis in the masters degree in pedagogics with specialisation in "Knowledge development and learning in working life".

Who is responsible for the research project?
The responsible for the research project is the Faculty of Educational Sciences at the University of Oslo.

Why are you asked to participate?
You are invited to participate as an employee of the software development company. The company is selected as a case because of its relevance to the research problem. The organisation is a true epistemic community, as the employees employ highly specialised knowledge in the collaborative process of developing the Odeon software.

What does participation imply for you?
Participation will imply that you take part in interviews about how the software is developed through various work processes and forms of collaboration, and how knowledge is shared in these processes.

UiO : University of Oslo

1/4

In order to gain insight into the mentioned work processes, we will conduct individual and group interviews, as well as meeting observations. Data will be gathered as electronic audio recordings, notes and relevant documentation on the work process (such as minutes, strategies and/or plans).

Participation is voluntary
It is voluntary to participate in the project. If you choose to participate, you can at any time withdraw your consent without giving a reason. All your personal information will then be made anonymous. Not participating or at a later stage withdrawing from the project will have no negative consequences for you.

Protecting your privacy - how we store and use your information
We will only use the data about you for the purposes described in this letter. We process the information confidentially and in accordance with data protection regulations, the General Data Protection Regulation and Personal Data Act (GDPR).

Only the master students conducting the study, Eirik Rise and Maja Greni Sundland, as well as the supervisor and professor Monika Bærøe Nerland, will have access to the data.

The audio recordings will be transcribed, and recordings, transcriptions and notes will be stored on the University of Oslo's secure servers. The data and company will be anonymized and will be unrecognisable in the publication.

To prevent unauthorised people gaining access to the personal information, names and contact information will be stored separately from other data.

What happens to your personal information after the research project is finished?
The personal information will be deleted when the project is finished, scheduled to be on the 31st of December 2022 at the latest.

What gives us the right to process personal information about you?
We process information about you based on your consent.

On behalf of the Faculty of Educational Sciences at the University of Oslo, NSD - Norsk senter for forskningsdata AS (Norwegian centre for research data) has assessed the handling of personal information in this research project to be in accordance with data protection regulations.

Your rights

As long as you can be identified in the gathered data, you have the right to:

- gain access to what information about you we are handling, and to receive a copy of the information
- to have wrong or misleading information about you corrected
- to have personal information about you deleted
- to send a complaint to Norwegian Data Protection Authorities about the handling of your personal information

If you have questions about the study, or wish to know more or make use of your rights, please contact:

- Faculty of Educational Sciences at the University of Oslo by
    - The master students Eirik Rise (email: eiriris@student.uv.uio.no or phone: +47 93871472) and Maja Greni Sundland (email: majagsu@student.uv.uio.no or phone: +47 99585128)
    - Professor and supervisor Monika Bærøe Nerland (email: m.b.nerland@iped.uio.no or phone: +47 22858172)
- Our Data Protection Officer Roger Markgraf-Bye (email: personvernombud@uio.no or phone: +47 90822826)

If you have questions about NSDs assessment of the project, please contact:

- NSD - Norsk senter for forskningsdata AS (Norwegian centre for research data) by email: personverntjenester@nsd.no or phone: +4753211500.

Yours sincerely,

Monika Bærøe Nerland	*Eirik Rise*	*Maja Greni Sundland*
Professor and supervisor	Student	Student

Please sign and return the consent form on the next page to majagsu@student.uv.uio.no

UiO : University of Oslo

C. Consent form

Consent form

I have received and understood information about the project *Knowledge sharing and knowledge development in expert communities – exploring collaborative work with joint epistemic objects* and

have had opportunities to ask questions.

I give consent to (check the appropriate boxes):

☐ participate in interviews
☐ participate in observations
☐ that my personal information is stored until the end of the project, until the 31st of December 2022

I give consent for my personal information to be processed until the project is finished 31st of December 2022.

Signed by project participant, date

Name in capital letters: ____

UiO : University of Oslo

D. Interview guide - individual interviews

Interview guide - individual interviews

Introductions

- Information about the project
- Your participation

Theme 1: Background

- Can you please shortly describe your educational background?
- What is your area of expertise?
- How long have you been working for the company?
- What are your main tasks and responsibilities in the company?

Theme 2: Team-organising

- What are you currently working on in the company? What is your main focus?
- How do you work together as a team?
  - How are you organised?
  - In what arenas does the work take place?
  - How do you divide responsibilities and tasks?
- What kind of team do you identify as? [Community of experts, professionals, etc.]
- ~~How do you identify as a team?~~
- How do you know what the others on the team are working on?
- What do you need from the others on the team to do your job?

Theme 3: Epistemic objects

- What parts of the software do you collaborate on developing currently?
  - What part is in the need of adoption / improvements?
- What tools or resources do you use for developing the software? [immediate objects, tools and artefacts]
- How do you use these tools?
  - What do they look like?
  - Can you share them with us?
- How do you document the development work?
  - Where do you document it? [notes, in the object, etc.]

1

- Do you work in line with some tendencies in the field of software development or engineering?
    - How would you describe your way of working?
    - How do you develop your way of working?

Sub-theme 3.1: Linking back to the observations / other interviews, focus on a specific tool or object identified to be especially relevant for knowledge development.

- How was it created / how did it come to be?
- Is it created based on needs from the customer, technical developments on the field, other things?
- How does the object structure / influence your work?
- How does it support knowledge sharing?

Theme 4: Knowledge sharing

- How does the team share knowledge in the company?
    - How do you organise the knowledge sharing process?
    - What does it look like?
- How do you coordinate your contributions in the software development?
    - In what ways do you use each other's knowledge?
- How do the tools assist the team in the knowledge sharing process?
- What challenges do you face when developing the software together?
    - How do you solve these issues together?
    - Can you describe such a process?
- ~~What is the role of the software itself in the process of developing new knowledge / functions together?~~
- How does the software push the team to develop new functions and knowledge?

Final questions

- Do you have anything to add? Anything you feel that you did not get a chance to say or were not reflected in the questions so far?

E. Interview guide - follow-up interview

Interview guide - follow-up interview

Theme 1: Origin

- Please describe the feature. What is it, what would you like us to know about it?
  - Can we see it / can you give us a demonstration?
  - Is it a new feature or an improvement of an already existing feature?
- Where did the idea come from?
  - How was it created? Is it created based on needs from the customer, technical developments on the field, other things?
  - Why did you develop this now?
- For how long have you been working on this feature?

Theme 2: Team collaboration

- What does the development process look like? Please describe it.
  - Who is involved in the process?
  - How do you make decisions?
  - What role has team-collaboration played in the process?

Theme 3: Knowledge sharing

- How do you share knowledge during the process?
  - What type of knowledge do you need to share?
  - How do you document the information?

Theme 4: Epistemic practices

- What are / have been the main challenges in developing this technology?
- Can you describe a problem you faced, that you couldn't solve alone, and how you went about solving it?
  - How did you work around these challenges?
  - What tools helped you in the process?

F. Interview guide - group interview

Interview guide - group interview

Theme 1: Introductions.

- How did you experience the interviews, were they as expected?
- How does a typical day at your company look like? Can you please describe it?

Theme 2: Team

- What are the strengths of you working as a team?
    - How do you complement each other on the team?
    - Are there any challenges when working as a team?
        - How do you come to an agreement?

Theme 3: Knowledge sharing

- What kind of knowledge do you need to develop the software?
- How do you share knowledge when developing the software?
- What challenges do you see when it comes to knowledge sharing on the team?
    - How do you overcome these challenges?

Theme 4: Epistemic objects

- How do you work together to solve complex problems you face when developing the software?
- What are the most important tools you use for developing the software and documenting the process?
- What drives the development of the software?
- When the company was founded there was an ambition to make the best room acoustics software available. What does the best room acoustics software look like?

G. Observation form (in Norwegian)

Observasjonsskjema

Hva er det som foregår her?

Arbeidsorganisering

- Hva skjer faktisk?
- Hvordan ser arbeidssituasjonen ut?
- Hvordan jobber de sammen?
- Hvordan fordeler de oppgaver og ansvar?
- Hvem gjør hva?

Dynamikk og relasjoner

- Hvordan ser prosessen ut?
- Hvordan fremstår relasjonene mellom teammedlemmene?
- Er det en hierarkisk eller flat struktur?
- Hvordan samordner de ulike ekspertiser?
- Hvem snakker når?

Arbeidsverktøy/objekter

- Hvilke verktøy er i bruk?
- Hvordan ser verktøyene ut?
- Hvilke måter dokumenterer de på under møtet?
- Hva utvikler de i programvaren?
- Hvilke funksjonaliteter jobber de med?

Kunnskapsdeling

- Hvilke problemer prøver de å løse?
- Hvilke utfordringer møter de i problemløsningen?
- Hvordan bruker de sin kunnskap for å løse problemene?
- Hvordan bruker de verktøy(ene) for å dele informasjon?
- Hvor henter de informasjon fra?