SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

Investigating Organizational Structures and Means for Effective Knowledge Sharing and Coordination in Large Agile Organizations

Anil Can Kara

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

Investigating Organizational Structures and Means for Effective Knowledge Sharing and Coordination in Large Agile Organizations

Untersuchung von Organisatorischen Strukturen und Mitteln für Effektiven Wissensaustausch und Koordination in großen agilen Unternehmen

Author:	Anil Can Kara
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Franziska Tobisch, M.Sc.
Submission Date:	15.05.2024

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15.05.2024 Anil Can Kara

Location, Submission Date Author

AI Assistant Usage Disclosure

Introduction

Performing work or conducting research at the Chair of Software Engineering for Business Information Systems (sebis) at TUM often entails dynamic and multi-faceted tasks. At sebis, we promote the responsible use of *AI Assistants* in the effective and efficient completion of such work. However, in the spirit of ethical and transparent research, we require all student researchers working with sebis to disclose their usage of such assistants.
For examples of correct and incorrect AI Assistant usage, please refer to the original, unabridged version of this form, located at this link.

Use of *AI Assistants* for Research Purposes

I have used AI Assistant(s) for the purposes of my research as part of this thesis.

Yes [x] No [ ]

Explanation:

- To get tips for certain LaTeX commands.
- To improve the writing of this thesis by using Grammarly.

I confirm in signing below, that I have reported all usage of AI Assistants for my research, and that the report is truthful and complete.

Munich, 15.05.2024	Anil Can Kara
Location, Date	Author

Acknowledgments

First and foremost, I would like to thank my advisor Franziska Tobisch for her great support throughout my thesis. Your constant motivation and constructive feedback were of great value.

In addition, I would like to thank all of our interview partners who contributed to this thesis by sharing their knowledge and experiences.

Furthermore, I would like to thank Professor Dr. Florian Matthes for making this research possible and giving me the opportunity to write my thesis at his chair of Software Engineering for Business Information Systems (SEBIS).

Last but not least, I would like to thank my family and all my friends who supported me throughout this great journey. Knowing that I can always count on your unconditional support is invaluable.

Abstract

Agile methodologies initially applied in individual software development teams at a small scale, have gained great popularity over the years. Many organizations seeking to stay up-to-date with ever-changing technological requirements and market dynamics have adopted agile ways of working to meet the needs of the business world and stay competitive. Although these methods were originally designed for small-scale teams, the proven success of such adoptions inspired many organizations to scale agile beyond individual teams so that their benefits can be reaped in larger contexts. However, due to their complex nature, the resulting large-scale agile software development environments can bring potential challenges regarding various practices, including knowledge sharing and coordination. Despite the structures that large-scale agile frameworks aim to provide to make the transition to larger scales smoother, they are still not sufficient to address the variety of challenges inherent to the large-scale agile software development contexts. Therefore, studying through which mechanisms knowledge sharing and coordination occur, which factors influence their efficiency, and in which contexts they appear in large agile organizations remains crucial. Although several case studies investigate knowledge sharing or coordination in large-scale agile settings, the existing literature is scarce in terms of going beyond individual organizations and providing a broader overview. To fill this gap, this thesis conducts a systematic literature review and an interview study and combines their findings to identify the relationships between various mechanisms, barriers, facilitators, and their application contexts. The systematic literature review provides the foundations, identifies the existing gaps within the topic, and draws certain conclusions to be assessed. Then, the interview study identifies a set of knowledge sharing and coordination mechanisms, a variety of barriers to and facilitators for their efficiency, and the contexts in which they tend to appear in large-scale agile environments. Further, it explores certain findings of the systematic literature review to assess their generalizability in practice. Then, the thesis outlines the key findings and provides a potential perspective for future research. The results indicate that knowledge sharing and coordination activities and the factors affecting them in large-scale agile organizations are tightly coupled, and the identified barriers and facilitators are interconnected.

Contents

1. Introduction

1.1. Motivation

Agile methodologies, which were initially applied in individual software development teams, have been quite successful and have gained great popularity over the years since the initial launch of Agile Manifesto [1] in 2001 [2] [3]. Many organizations have adopted an agile way of working in their teams to be able to stay competitive in the market where customer needs, technological requirements, and the market itself change rapidly [4] [5]. Even though the agile methodologies were designed for small-scale teams [6] [7] [8], the proven success of such adoptions inspired organizations to scale agile principles beyond individual teams so that their benefits can be reaped in larger contexts [9]. Some large-scale Agile frameworks have been introduced, such as SAFe [10] and LeSS [11], to ensure that the organizations aiming to scale agile methodologies to a larger extent can follow a structured approach with prescribed roles, rules, and best practices [10] [11]. Regardless of whether these frameworks are applied or not, scaling agile beyond individual teams has brought a variety of challenges to organizations [12]. Due to the complex nature of communication between multiple teams working towards a united goal and the potential introduction of new organizational structures bringing people with different backgrounds and skill sets together, maintaining an efficient way of knowledge sharing and coordination among the cross-functional team members emerges as a challenge in front of the successful application of agile practices at large-scale, even if they were previously implemented at the team level seamlessly. Therefore, investigating the means for effective knowledge sharing and coordination in a large-scale context is essential.

Existing literature covers the topic of knowledge exchange and coordination in large-scale agile environments by mainly focusing on individual case studies in which a single organization is investigated in detail. While these studies provide a solid background regarding the research area, their results face certain validity threats due to the characteristics of case studies in general [13]. As a result, drawing generalizable conclusions becomes infeasible. In addition, some such studies concentrate solely on a specific aspect of the topic, such as Scrum-of-Scrums [14], rather than elaborating on what means and organizational structures are used to realize effective knowledge sharing and coordination. Thus, an extensive overview of the mechanisms used for knowledge exchange and coordination, as well as the hindering and facilitating factors for them in large-scale agile settings is lacking in the literature. This thesis aims to fill this research gap by shedding light on how knowledge sharing and coordination take place in large-scale agile organizations, which organizational structures and means are available for effective knowledge sharing and coordination, what barriers and facilitators are observed in such settings, as well as the benefits, trade-offs, and application contexts of the

mechanisms.

1.2. Research Questions

In light of the above-mentioned motivation and research goals, this thesis is organized to answer the following three main research questions:

> Research Question 1: How do knowledge exchange and coordination take place in large agile organizations, and which mechanisms are used for this purpose?

The first research question addresses the importance of knowledge exchange and coordination fundamentals by inquiring how they occur in large-scale agile setups. This is particularly essential as answering it provides the first steps towards achieving a broad overview of the mechanisms in use.

> Research Question 2: What are the barriers to and facilitators for effective knowledge sharing and coordination in large agile organizations?

Scaling successfully applied agile practices to a larger extent brings a diverse set of challenges in terms of knowledge sharing and coordination [12]. In order to gain a deeper understanding of these challenges as well as what factors might worsen or mitigate them, the second research question centers upon the barriers in front of efficient coordination and knowledge sharing in large-scale agile organizations, together with the potential facilitators that enable efficient knowledge sharing and coordination.

> Research Question 3: What are the benefits, trade-offs, and application contexts of knowledge exchange and coordination mechanisms in large agile organizations?

Finally, the third research question investigates what potential benefits and problems each mechanism potentially brings, and which mechanisms are useful in which contexts to enable effective knowledge sharing and coordination in large-scale agile organizations. Answering this research question is critical to understand the application areas of the available means and to put them into a certain context.

1.3. Research Approach

To answer the above-mentioned research questions, this thesis combines two research approaches: a systematic literature review and a set of semi-structured expert interviews.

A systematic literature review is a scientific methodology for identifying and assessing all existing research related to a particular topic. Applying it thoroughly enables researchers to spot research gaps in their topic of interest and to draw fair conclusions about their research area [15]. For this thesis, besides the conclusions drawn from the systematic literature review, it is also important to identify the gaps in the existing research to design the interview study in a way that helps to fill those gaps. Further details about the systematic literature review approach undertaken in this thesis are discussed in Chapter 4.

In the second part of this study, semi-structured expert interviews are conducted. The term semi-structured refers to the nature of the interviews in which the interview partners are asked a set of predefined open-ended questions throughout each session, but the order and the exact wording of the questions are flexible, and additional follow-up questions are possible depending on the answers received [16]. These interviews search for answers to the research questions further by getting practical insights from the experts in the field without omitting the obtained results and the identified gaps during the systematic literature review. In this way, the two parts of the study are aimed to complement each other by combining the theoretical foundations and the practical insights. Further details about the semi-structured interviews are presented in Chapter 5.

1.4. Structure of the Thesis

The outline of the remainder of this thesis is as follows: Chapter 2 provides a brief overview of the concepts that are relevant to the thesis including the introduction of agile, agile frameworks, scaling agile, large-scale agile frameworks, knowledge sharing, and coordination. Chapter 3 introduces some of the relevant publications in the literature that are related to the objectives of this thesis. Chapter 4 explains how the systematic literature review is conducted, while Chapter 5 demonstrates the steps of the interview study and provides descriptive study data. By combining the findings from the systematic literature review and the interview study analysis, Chapter 6 presents the results of this thesis including the identified mechanisms, barriers, and facilitators for effective knowledge sharing and coordination in large agile organizations. Chapter 7 summarizes the key findings of the thesis, discusses the results, and reflects on the key limitations. Finally, Chapter 8 provides a brief summary and closing remarks for future research.

2. Foundations

This chapter establishes the groundwork for this thesis by introducing the fundamental concepts that are highly relevant to this study. Section 2.1 describes the emergence of the agile methodology and the Scrum framework, while Section 2.2 focuses on the attempts to scale agile and the associated challenges. In Section 2.3 and Section 2.4, knowledge sharing and coordination in non-agile, agile, and large-scale agile settings are described.

2.1. Agile Methodology

Traditional software development methods rely on thorough planning, prior specifications, and heavyweight processes [17]. In today's software development environments, these methods are highly challenged by fast-changing ecosystems and requirements for successful software development practices [18]. On the contrary, agile software development methodologies offer a lighter-weight, faster, and nimbler software development process [19]. They follow an incremental delivery approach and aim to reduce overhead tasks and planning efforts to cope with the unpredictability inherent to the software development process [20].

The first agile initiatives began in the nineties with the introduction of several lightweight frameworks in software corporations [21]. Since then, many organizations have adopted agile approaches, especially after the declaration of the Agile Manifesto [1] in 2001. This section describes the Agile Manifesto and the emerging agile methodologies and frameworks. Upcoming sections describe the attempts to scale them to larger contexts, define knowledge sharing and coordination, and their applications in large-scale agile environments.

2.1.1. Agile Manifesto

In 2001, a group of twelve people officially declared the Agile Manifesto containing four values and twelve principles [1]. These fundamental values and principles are demonstrated in Table 2.1 and Table 2.2. As the keystones of the agile methodologies, these values and principles will be referred to throughout this thesis to establish connections between knowledge sharing and coordination dynamics and the agile values and principles.

2.1.2. Scrum

In their publication in 1997, Schwaber [22] introduced Scrum as a development process that increases flexibility and responsiveness to the ever-changing requirements of development [22]. Later, they extended the methodology and provided specific insights for the domains of

Values in the Agile Manifesto	
Value 1	Individuals and interactions over processes and tools
Value 2	Working software over comprehensive documentation
Value 3	Customer collaboration over contract negotiation
Value 4	Responding to change over following a plan

Table 2.1.: Values described by the Agile Manifesto [1]

Principles behind the Agile Manifesto	
Principle 1	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
Principle 2	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
Principle 3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
Principle 4	Business people and developers must work together daily throughout the project.
Principle 5	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
Principle 6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
Principle 7	Working software is the primary measure of progress.
Principle 8	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
Principle 9	Continuous attention to technical excellence and good design enhances agility.
Principle 10	Simplicity – the art of maximizing the amount of work not done – is essential.
Principle 11	The best architectures, requirements, and designs emerge from self-organizing teams.
Principle 12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Table 2.2.: Principles behind the Agile Manifesto [1]

software development [23] and project management [24], and published extensive guidelines [25] to define the borders of the methodology.

In Scrum, development is split into iterations called Sprints, typically lasting between two to four weeks [25]. Sprints consist of four main events, namely Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective [25]. Each Sprint starts with Sprint Planning in which the tasks of the upcoming Sprint are discussed and decided jointly [25]. During the Daily Scrum, which is a 15-minute event that takes place every day at the same time slot, team members update each other regarding their progress and assess their current state toward the product goal [25]. In the Sprint Reviews, the Scrum team reflects on the Sprint result [25]. Finally, Sprint Retrospective ends each Sprint and aims to look for ways to improve efficiency for the upcoming Sprints [25].

2.2. Scaling Agile

The success of the agile methodologies gave rise to many attempts to use them in larger contexts [9]. There is no unified definition of what large-scale exactly refers to, but experts define large-scale by using a variety of different measures such as the number of developers, the number of time zones, the size of the codebase, the number of teams, and many others [26]. Throughout this thesis, we adhere to the definition that describes large-scale software development as a development process in which multiple teams work on the same products [26]. Although Dingsøyr et al. [27] further introduces the term very large-scale development as a practice involving more than ten teams, our study does not distinguish between the two. This section introduces popular large-scale agile frameworks, their main characteristics, and the challenges associated with scaling practices.

2.2.1. Large-Scale Agile Frameworks

Inspired by the success of the small-scale implementation of agile methodologies, large-scale agile frameworks such as SAFe, LeSS, and others aim to scale the benefits of team-specific agile practices to larger contexts [10] [11] [28] [29]. This section briefly outlines some of the most popular large-scale agile frameworks that are relevant for the remainder of this thesis.

Scaled Agile Framework (SAFe)

Scaled Agile Framework (SAFe), initially released in 2011 [30], is a system that enables teams of teams, business units, and entire organizations to adopt agile at a larger scale [10]. It builds on the basic agile principles and values [1] as well as the principles of Lean Thinking by providing specific guidance for scaling in large, complex, distributed, or high-compliance settings [10]. By doing so, SAFe extends existing agile principles and values from the individual teams to the teams of agile teams and portfolios [10]. Like the Agile Manifesto [1], SAFe also provides ten immutable Lean-Agile principles to inspire and inform the roles

and practices prescribed by the framework [31]. Some of these principles are referred to in the remainder of this thesis with their numbers as specified in Table 2.3.

SAFe Lean-Agile Principles	
Principle 1	Take an economic view.
Principle 2	Apply systems thinking.
Principle 3	Assume variability, preserve options.
Principle 4	Build incrementally with fast, integrated learning cycles.
Principle 5	Base milestones on objective evaluation of working systems.
Principle 6	Make value flow without interruptions.
Principle 7	Apply cadence, synchronize with cross-domain planning.
Principle 8	Unlock the intrinsic motivation of knowledge workers.
Principle 9	Decentralize decision making.
Principle 10	Organize around value.

Table 2.3.: Principles of SAFe [31]

Designed to work in conjunction with the principles outlined, SAFe describes four different configurations that are suitable for different organizations: Essential SAFe, Large Solution SAFe, Portfolio SAFe, and Full SAFe [32]. Essential configuration describes two flows, namely the Agile Release Train (ART) Flow that is governed by Product Managers, System Architects, and Release Train Engineers, and the Team Flow that is handled by the agile teams [32]. ART refers to a team of agile teams that develops and delivers incrementally [33]. Solution Train describes an organizational structure that is used to construct large solutions and often contains multiple ARTs [34]. Large Solution SAFe extends the essential configuration by introducing the Solution Train Flow, which is under the responsibility of the Solution Managers, Solution Architects, and Solution Train Engineers. Portfolio configuration replaces the Solution Flow in the Large Solution SAFe with Portfolio Flow handled by the Epic Owners and the Enterprise Architects [32]. Full SAFe keeps both the Solution Train Flow and the Portfolio Flow in the configuration [32].

Large-Scale Scrum (LeSS)

LeSS is defined as a scaled version of one-team Scrum, maintaining most of the ideas in one-team Scrum [11]. It is designed for up to eight teams of eight people each that work on a single product backlog [11]. In LeSS, one Product Owner is delegated to maintain the entire product backlog, and all teams share a common Sprint [11]. LeSS differentiates itself from one-team Scrum mainly by encouraging an overall Product Backlog Refinement, an overall Retrospective, observation of other teams' Daily Scrums, and establishing a Sprint Planning with two phases, including all teams in one phase and holding individual plannings in the other.

The Spotify Model

The Spotify Model, published in 2012, is a scaled agile methodology used at Spotify to make agile teams in multiple locations work harmoniously on a single product [28]. It introduces the notion of Squads which are similar to the Scrum Teams, and the concept of Tribes that are made of Squads [28]. In addition, it suggests the concept of Guilds, consisting of individuals with a certain interest across all the Tribes [28]. In each Tribe, members that share a common set of skills and specialties constitute Chapters [28].

Scrum@Scale

Scrum@Scale is a lightweight organizational framework that facilitates a network of teams operating with the Scrum Guide to address complex problems while creating and delivering products with optimal value [29]. Scrum@Scale does not restrict the product type and aims to work equally well with digital, physical, or any other type of product [29]. In Scrum@Scale, each team contains a peculiar Scrum Master and a Product Owner. It introduces the scaled versions of the classical Scrum events, such as The Scaled Daily Scrum, and The Scaled Retrospective, to orchestrate how each Sprint works [29]. Similarly, some variants of the roles in Scrum are integrated into the framework to handle broader responsibilities, such as The Chief Product Owner, who is responsible for coordinating the priorities in the Product Owner Team and managing The Scaled Backlog Refinement and The Scaled Sprint Review [29].

Scrum-of-Scrums

Scrum-of-Scrums, or Meta-Scrum, is a construct that brings individuals from different Scrum Teams together to collaborate and work towards the same product goals [35].

2.2.2. Challenges

Scaling agile to large-scale contexts is not a straightforward endeavor and brings various challenges [36]. Uludağ et al. [36] conduct a structured literature review and classify the arising challenges into eleven categories, including communication and coordination,

knowledge management, tooling, geographical distribution, and software architecture. The next subheadings briefly describe the most relevant categories to this thesis.

Communication and Coordination

Challenges arising in large-scale agile development environments regarding communication and coordination include but are not limited to the coordination of multiple agile teams working on the same product, establishing self-organization, and dealing with increased efforts by establishing inter-team communication [36]. In addition, several other challenges classified under different categories also directly relate to the coordination of agile teams, such as coordinating geographically distributed agile teams and facilitating agile teams to participate in cross-shore meetings, which are classified under the category geographical distribution [36].

Knowledge Management

Facilitating shared context and knowledge, sharing a common vision, creating lightweight documentation, establishing a common scope for different stakeholder groups, and dealing with internal silos are classified as the challenges arising in large-scale agile development contexts regarding knowledge management [36]. Together with the above-mentioned coordination-related challenges, this thesis investigates these challenges in more depth to identify the contexts in which they are more likely to appear, the barriers causing them, and their potential solutions by certain facilitating practices.

2.3. Knowledge Sharing

Knowledge is widely perceived as an organization's most strategic resource, and its management is seen as a critical factor for an organization's success [37], and even as a significant predictor of business performance [38]. Therefore, many researchers have investigated knowledge sharing in various contexts to identify how it occurs, and in which ways it is enhanced or deteriorated [37] [38] [39]. This section defines knowledge sharing and highlights the related characteristics in agile and large-scale agile environments.

2.3.1. Definition

As a general notion, Helmstädter [40] defines knowledge sharing as the interaction between human actors where the material is knowledge. It is mainly a natural process that occurs automatically [38] and spontaneously [41], and it is subject to changes on an individual level [38]. In the context of firms, knowledge sharing can be defined as the transfer of skills, technology, and wisdom between the units of an organization [42]. It also represents the collective beliefs or routines related to exchanging employee knowledge, experiences, and skills across the organization [43]. It can take place between the individuals, within and among the teams, within and among the organizational units, and within and among

organizations [44]. Due to its variety of application contexts, it is a complex phenomenon with many different aspects to investigate. This thesis focuses on its application in large-scale agile contexts. To establish the foundations for the remainder of this thesis, the next subsections demonstrate the groundwork and definitions in both agile and large-scale agile environments.

2.3.2. Knowledge Sharing in Agile

As outlined in Table 2.1, four values of the Agile Manifesto propose a new, agility-centered way of prioritization compared to the traditional development approaches [1]. As an activity that can occur at every organizational level [44], knowledge sharing is highly affected by such a paradigm shift. By applying the reformative values and principles of the Agile Manifesto, agile teams practice new mechanisms of sharing knowledge within their teams [25]. Therefore, studying knowledge exchange in agile settings is crucial for understanding its influence in larger contexts.

Agile methodologies support knowledge sharing within the team with various practices. For instance, pair programming, release planning, iteration planning, and pair rotation help team members to share their knowledge and expertise with their colleagues in Extreme Programming (XP), while dailies, cross-functional teams, and retrospectives facilitate this in Scrum [45, 46, 47]. However, the mode of knowledge sharing varies among different practices. For instance, while Daily Scrum meetings facilitate knowledge sharing within the entire team, pair programming facilitates knowledge sharing within a pair of developers [47]. In addition, these practices do not suggest any specific knowledge sharing mechanism at organizational levels.

2.3.3. Knowledge Sharing in Large-Scale Agile

As briefly described in Section 2.3.2, agile methodologies propose various mechanisms to share knowledge effectively within the teams [25]. However, they do not offer a specific solution for inter-team knowledge sharing [48]. As a result, the efficiency of the knowledge sharing practices is highly challenged in large-scale settings because knowledge sharing within the team is significantly easier than with colleagues beyond the team [49]. This is particularly relevant for large-scale agile organizations where multiple teams must share knowledge and align on the same goals to achieve their business goals.

Large-scale agile organizations attempt to tackle the problem of inter-team knowledge sharing by introducing various mechanisms such as Scrum-of-Scrums, Communities of Practice, and project member rotations [48] [49]. However, none of these mechanisms are challenge-free [48] [49]. In addition, informal conversations and regular meetings still constitute a large part of the knowledge sharing events even with the existence of explicit mechanisms for inter-team communication [49]. This thesis explores how knowledge sharing takes place, which barriers deteriorate its efficiency, and which facilitators enable more effective knowledge sharing practices in large agile organizations.

2.4. Coordination

This section defines coordination both as a general notion and from an organizational point of view and highlights its relevant characteristics in agile and large-scale agile environments.

2.4.1. Definition

From a broad perspective, Bernstein [50] defines coordination as a problem of mastering a very large number of degrees of freedom involved in a particular movement by reducing the number of independent variables to be controlled [50]. Similarly, Mooney [51] describes coordination as the methodical arrangement of a group effort to provide a unity of action in the pursuit of a common purpose [51]. Even though these definitions are highly general, they still contain the characteristics of organizational coordination in the sense that they emphasize the importance of common goals and reducing obstacles.

From an organizational point of view, Mooney [51] further defines coordination as the first principle of an organization [51]. They suggest that the other organizational principles serve only as the means for facilitating coordination [51]. For Van de Ven et al. [52], coordination is the activity that involves linking different parts of an organization to carry out a certain set of tasks [52]. Based on the theory of organizations, they claim that the need for coordination is fundamental to the organizations [52]. Camerer and Knez [53] argue that the need for coordination arises when the success of an organization depends on the decisions taken by the groups of individuals that make up the organization [53].In today's world, where companies can operate even at global scales, the decision-makers have the power to highly impact an organization's future, and the emergence of the need for coordination is therefore unavoidable. As a result, the topic of coordination shows itself as a crucial field of study for researchers studying the dynamics of organizations. Even before the agile methodologies came into sight, coordination was considered a critical element of organizations and software development practices [52] [54] [55]. To establish the foundations for the topic of this thesis, the next subsections describe how coordination is handled in the contexts of agile and large-scale environments.

2.4.2. Coordination in Agile

Similar to the reasons underlined for knowledge sharing, the way coordination occurs is also highly subject to the principles and values suggested by the Agile Manifesto [1]. Value 1 in the Agile Manifesto, as depicted in Table 2.1, suggests prioritizing interactions over processes and tools. Principles number 4, 5, 6, 11, and 12 listed in Table 2.2 provide direct or indirect guidance about how coordination should take place, and prescribe the coordination of business people and developers, face-to-face communication, self-organization, and self-reflection for practicing effective coordination [1].

Agile methodologies attempt to facilitate coordination in agile teams through various practices, such as Daily Standups, Sprint Reviews, and Sprint Retrospectives in the Scrum framework

[25], and many others depending on the chosen agile framework. Similar to the previously mentioned knowledge sharing mechanisms, the majority of the agile methodologies do not explicitly describe how inter-team coordination should occur, even though certain constructs such as Scrum-of-Scrums are proposed by the Scrum framework [14] [35].

2.4.3. Coordination in Large-Scale Agile

Large agile organizations use various mechanisms to coordinate beyond the team level. These mechanisms include but are not limited to the implementation of agile processes within the organization, usage of documentation and planning tools, and open work areas [56]. Similar to the knowledge sharing mechanisms, none of these practices are challenge-free, and the informal mechanisms such as ad-hoc meetings as well as the regular meetings like Daily Standups still constitute an important part of the coordination efforts in large-scale agile organizations, even if they are not introduced explicitly to facilitate inter-team coordination [56]. This thesis investigates these mechanisms to elaborate more on how coordination takes place and identifies the associated barriers and facilitators for effective coordination in large-scale agile environments.

3. Related Work

This chapter outlines some of the existing publications that are highly relevant to this thesis and their contributions to the different aspects of the topic.

Paasivaara et al. [14]

Paasivaara et al. [14] study the topic of inter-team coordination in large-scale Scrum environments by focusing on a specific type of meeting, Scrum-of-Scrums. As they emphasize, it is the only mechanism proposed by the Scrum methodology for inter-team coordination [35]. Based on two case projects and 58 expert interviews with product owners, architects, developers, testers, and managers, they argue that Scrum-of-Scrums severely fails to meet the needs for effective coordination in large-scale Scrum projects due to the wide audience, in which keeping everybody interested is highly challenging [14]. They further demonstrate that the meeting participants mostly did not know what information was valuable for the other teams, and therefore, they usually ended up not reporting anything in these meetings [14]. To tackle these problems, one of the projects introduced feature-specific Scrum-of-Scrums meeting which was observed to work better according to the researchers [14].

Bick et al. [17]

Bick et al. [17] investigate a specific type of coordination challenge emerging in large-scale software development teams with a case study conducted at a global enterprise software company [17]. In their research, they specifically focus on the hybrid approaches that apply traditional coordination strategies between the teams and agile methodologies within the teams [17]. Based on 23 semi-structured expert interviews conducted with product owners, scrum masters, a chief product officer, and a lead architect, they argue that ineffective coordination results from the lack of dependency awareness among the team members [17]. Further, they propose that dependency awareness can be achieved by aligning planning activities such as specification, prioritization, estimation, and allocation, resulting in more effective coordination [17]. However, their research does not include various other challenges that might impact coordination in similar contexts.

Santos et al. [48]

Santos et al. [48] focus on inter-team knowledge sharing effectiveness in agile software development organizations [48]. By using grounded theory and analyzing the data collected from four Brazilian organizations and one agile expert, they propose a conceptual model that explains how knowledge sharing between agile teams is related to purposeful practices and

organizational conditions [48]. The resulting model demonstrates that effective inter-team knowledge sharing in large-scale agile contexts cannot be achieved solely by applying a list of defined approaches, rather it is important to create a sustainable context where the entire process is implemented agile [48].

Stray at al. [57]

Stray et al. [57] investigate the topic from a specific perspective by narrowing it down to the usage of virtual communication tools for coordination and knowledge sharing in large-scale agile organizations [57]. For this purpose, they conduct a case study at a large software company producing software for the engineering field [57]. From 30 team members who are distributed across different sites, they collect chat logs that amount to roughly 30,000 messages between the team members from the virtual communication tool Slack[1] [57]. In addition, they conduct some unstructured interviews and observe some of the retrospective meetings at the organization [57]. By analyzing the data, they identify challenges such as language differences, unbalanced activity, too much usage of direct messaging, and insufficient setup of the channels [57]. Moreover, they also demonstrate that Slack facilitates building team awareness, which is significantly relevant for dependency awareness introduced by Bick et al. [17] as a necessity for effective coordination [57]. Even though the study provides relevant insights about our research topic, it is only limited to one tool, Slack, and therefore, lacks a broader overview of virtual communication practices in general.

Dorairaj et al. [58]

Dorairaj et al. [58] describe how Agile teams collect, store, share, and use knowledge in the context of distributed software development teams [58]. Based on 45 expert interviews from 28 different companies in the USA, India, and Australia, using the grounded theory, they categorize the factors that promote effective knowledge management under four main categories, namely knowledge generation, knowledge codification, knowledge transfer, and knowledge application [58]. For each category, they provide the facilitating factors for effective knowledge management [58], which is highly relevant to the second research question of our study, as described in Chapter 1.

Berntzen et al. [59]

Berntzen et al. [59] conduct a case study with 31 interviews and 25 participants consisting of product owners, program managers, program architects, tech leads, and team leaders [59]. Additionally, they collect Slack logs, Confluence documentation, and e-mails, and they observe 9 different types of meetings on-site [59]. Using thematic analysis on the large data obtained, firstly, they identify a total of 27 coordination mechanisms and categorize them under three main categories, namely meetings, roles and artefacts, and tools [59]. Further, they divide each category into two and come up with a taxonomy with six classes as

[1]https://slack.com/

follows: scheduled/unscheduled meetings, individual/team roles, tangible/intangible tools and artefacts [59]. Secondly, they demonstrate that each coordination mechanism display a combination of four key characteristics, namely technical, organizational, physical, and social [59]. The results of the study provide a structured way of investigating coordination mechanisms in large-scale agile organizations and an extensive list of such mechanisms with their characteristics, which are important to understand the trade-offs and the applications contexts of those mechanisms.

Yang & Wu [41]

In their study, Yang & Wu [41] investigate knowledge sharing as a general phenomenon without restricting it to any specific type of organization [41]. Their approach differs significantly from other publications outlined in this chapter in the sense that their research relies on simulating knowledge sharing in an organization rather than conducting a field study [41]. Besides demonstrating that knowledge sharing occurs spontaneously most of the time, their results also show that regardless of the strategy followed during knowledge sharing practices, the initial states of the actors are crucial for the overall knowledge sharing process [41]. In addition, they propose a rewarding scheme that incentivizes knowledge sharing in organizations and conclude that rewarding individual knowledge sharing actions is more effective than periodic rewards adopted at the organizational level [41]. Although their research does not specifically focus on agile environments, their results are highly relevant for this thesis to shed light on the factors that facilitate knowledge sharing in such organizations.

Ghobadi & Mathiassen [60]

Ghobadi & Mathiassen [60] study the barriers to effective knowledge sharing in agile software teams by considering the perspectives of the individuals [60]. Even though their study does not focus on large-scale knowledge sharing, it elaborates on how barriers are perceived by the relevant parties of the knowledge sharing process and how the variance of these perceptions can be handled effectively [60]. Their results suggest that project managers primarily value removing project setting barriers, while developers, testers, and user representatives are interested in mitigating communication, organization, and team capability-related barriers [60]. Additionally, they identify 37 barriers, categorize them across 7 types, and argue that it is essential to take every party's concerns into consideration in order to achieve effective knowledge sharing among the team members [60]. These results are relevant to our study to understand the dynamics of the barriers to knowledge sharing in larger agile settings, as different perceptions of the same phenomenon can take place on the inter-team level with some similar characteristics to the intra-team level.

4. Literature Review

This part of the study follows the literature review procedures suggested by Kitchenham [61]. As the first step, several digital databases to be queried have been identified to cover the existing literature as comprehensively as possible. With their rich query options and publications, Scopus[1], IEEE Xplore[2], and ACM Digital Library[3] were selected as the main sources of publications for the literature review part of the study. In addition, Google Scholar[4] has been used to find several complementary publications that do not exist in the above-mentioned databases. However, it is not used as one of the main sources, as the querying and filtering options are highly limited compared to the others. One approach might have been to consider only the top publications returned by Google Scholar, but the sorting algorithm behind it is not fully known. Thus, it was not applicable to the systematic literature review approach followed in this thesis.

Secondly, keywords have been identified to develop a search strategy that yields as many relevant publications as possible. After an initial examination of the databases, the following keywords have been observed to be the ones that are used the most to refer to large-scale contexts: *large, large-scale, scaling, scaled, inter-team, multiteam, distributed*. It is important to note that using the keyword *distributed* in the search string yields many studies focused on distributed software development. However, since the distributed software development environments can substatially vary in the setup and the size, it is not possible to classify all of the resulting publications as relevant for large-scale without assessing them individually. Therefore, some publications that do not involve large-scale characteristics have been eliminated after careful consideration. Still, the keyword has been kept in the search string to obtain the relevant studies. The remainder of the keywords did not have the same issue and returned large-scale relevant publications.

Similarly, the set of keywords to be included in the search string for knowledge sharing and coordination have been selected as *knowledge sharing, knowledge exchange, knowledge management, coordination*, and for agile contexts as *agile, scrum*. Even though the Scrum framework is a subset of agile in general, it is by far the most used agile framework in the industry for software development [62]. Thus, it is included in the search string to avoid missing any publications that possibly mention Scrum instead of explicitly using the word agile.

[1]https://www.scopus.com
[2]https://ieeexplore.ieee.org
[3]https://dl.acm.org
[4]https://scholar.google.com

Finally, inclusion and exclusion criteria have been identified to keep the search results relevant to the study's main goals. Since the Agile Manifesto was published in 2001, only the studies after 2001 have been included. In addition, several other criteria have been selected regarding the language and the accessibility of the papers. Table 4.1 summarizes the inclusion and exclusion criteria followed in this literature review.

Inclusion Criteria
• Search string matching the title or the keywords
• Full text is accessible online
• Published after 2001
• Written in English
Exclusion Criteria
• Content not relevant to answer the research question
• Duplicates

Table 4.1.: Inclusion and exclusion criteria of the systematic literature review

The above-mentioned process of search strategy formulation resulted in the following final search string, although the exact syntax of it differs in different databases:

> ("agile" OR "scrum") AND ("large" OR "large scale" OR "large-scale" OR "scaling" OR "scaled" OR "inter-team" OR "multiteam" OR "distributed") AND ("knowledge sharing" OR "knowledge exchange" OR "knowledge management" OR "coordination")

The queries were run with the search string above on the titles and the keywords of the publications in Scopus, IEEE Xplore, and ACM Digital Library. The keyword sections constructed by the authors and the publications' titles are observed to work well in determining if a study is relevant to our search string. As a result, abstracts are omitted during the search because of the high number of occurrences of widely used words such as *coordination* even when the paper's focus is completely different than what this thesis studies. After the removal of duplicates, 102 publications were gathered in total, which was reduced to 59 after eliminating the papers that did not fulfill the relevance criteria. The exact number of results from each database is summarized in Table 4.2.

Database	Query Hits	Inclusion Criteria Not Fulfilled	Remaining
Scopus	91	2	89
IEEE Xplore	26	0	26
ACM Digital Library	44	1	43
Subtotal			**158**
Exclusion Criteria			
- Duplicate removal			-56
- Content not relevant			-43
Total			**59**

Table 4.2.: Query results of the systematic literature review

5. Interview Study

Conducting interviews is one of the most important data collection methodologies in qualitative research [16]. In this chapter, the interview study, which is conducted to obtain a broad overview of the means, organizational structures, barriers, and facilitators for effective knowledge sharing and coordination in large-scale agile organizations is presented. Section 5.1 describes the main objectives of the study. Section 5.2 demonstrates the steps taken during the preparation phase before conducting the interviews and the questions that are asked to the interview partners in order to find answers to the research questions described in Chapter 1. In Section 5.3, the interviews and the process of data collection are described. Finally, Section 5.4 demonstrates how data analysis is performed.

5.1. Study Objectives

The main objective of the interview study is to portray the organizational structures and mechanisms used in the industry for coordinating and sharing knowledge effectively, as well as facilitating and hindering factors accompanying them in large-scale agile contexts. Practical insights collected during the interview study are treated as a complementary part of the knowledge obtained from the literature review in the search of answering our research questions.

5.2. Study Preparation

Throughout this interview study, a semi-structured approach in which the interviewer has the opportunity to modify the order and the wording of the questions has been followed [16], as described in Chapter 1. In a study aiming to shed light on the broad overview of mechanisms used, this approach is particularly useful to obtain the best possible results as it gives the freedom needed for the interview partners to provide their experiences in an extensive perspective without being fully restricted to a strict interview process. In addition, having the opportunity to include follow-up questions in the interviews is essentially beneficial while interviewing experts with various backgrounds and experiences so that their unique perspectives on the topics can be explored more effectively.

In the light of this semi-structured nature, a set of key points have been identified to guide the interview partners towards the right direction, i.e., to keep their answers relevant to our research questions. The key points that are used in formulating the interview questions are as follows:

- Role(s) undertaken by the interview partner in large-scale agile environments
- Years of experience of the interview partner
- Defined roles and the teams' setup in the large-scale agile context
- Size of the teams and the organization
- Product that the agile teams in question are/were working on
- Large-scale agile frameworks used in the organization, if any
- Tools used for knowledge sharing and coordination
- Knowledge sharing and coordination mechanisms used within the organization
- Observed barriers in front of effective knowledge sharing and coordination
- Structure of the organization and its effect on knowledge sharing and coordination, if any
- Facilitating factors experienced for effective knowledge sharing and coordination
- Trade-offs of the knowledge sharing and coordination mechanisms used in large-scale agile organizations
- Interviewees' suggestions for enabling effective knowledge sharing and coordination, or mitigating the effect of the observed challenges

Based on the initial findings obtained from the literature review, the interview partners are also asked about the following topics to validate if the challenges and facilitators outlined by the case studies in the existing literature have any generalizability:

- Pros and cons of working in a co-located or a remote setting regarding knowledge sharing and coordination
- Role of ad-hoc meetings and their comparison to scheduled meetings regarding knowledge sharing and coordination
- Dependencies arising in large-scale agile environments and ways to cope with them
- Efficient setup of communication tools and their relevance to knowledge sharing and coordination
- Documentation efforts to enable effective knowledge sharing and coordination

In order to find answers to our research questions successfully, focusing on the above-mentioned points was considered necessary but not sufficient. Depending on the initial answers received about the key points described, every interview partner has been steered into different directions with certain follow-up questions that are related to their unique contexts, so that they can enrich the conversation with the topics they are familiar with. An outline of the questions asked during the interview study is listed in the Appendix.

The selection of the interview participants was an important part of the preparation process. Since the topic of knowledge sharing and coordination is a relevant concept for any type of role in any sector, participants have been chosen in a way that the entire set covers as many roles and industries as possible to ensure a variety of voices. In this way, the goal is to avoid having one-sided views on the investigated topics, which would undermine the purpose of achieving a broad view. The participants of the study had the roles outlined in Table 5.2.

5.3. Data Collection

A total of 11 experts with various levels of experience in large-scale agile contexts have been interviewed in 10 sessions throughout the study, the first on 5 February 2024 and the last on 19 April 2024. All of the interviews were conducted remotely on the online video conferencing tool Zoom[1] and recorded with the consent of each participant for transcription purposes. The average duration of the sessions was 55 minutes, the shortest one lasting 42 minutes and the longest one lasting 67 minutes, excluding the introductory parts in which the interviewees were briefly introduced to the research goals and the interview structure. All interviews were conducted in English.

5.4. Data Analysis

This section describes the steps performed after the data collection, which are transcription, coding, and analysis of the interviews.

5.4.1. Transcription

Based on the recordings collected from each session, transcriptions are prepared and made ready for analysis by coding. The coding and the analysis phases are explained in Section 5.4.2. During the transcription phase, any confidential information is kept hidden to ensure anonymity. All transcriptions are created and stored in the qualitative data analysis tool MAXQDA[2].

5.4.2. Coding and Analysis

We coded the interview transcripts by following the two-cyclic approach proposed by Saldaña [63]. During the first cycle, we followed the descriptive coding approach, where the text segments are summarized with a short phrase explaining the content in the text chunk [63]. After the first cycle was complete, we used the pattern coding in the second cycle to group similar codes under certain categories [63]. The overall coding process included both deductive and inductive coding, as we initially started with pre-defined codes originating from our research questions but also defined new codes inspired by the content of the

[1]https://zoom.us
[2]https://www.maxqda.com

interview transcripts during the coding phase [64]. We used MAXQDA to perform all the codings.

The above-mentioned coding process resulted in seven first-level codes, and 158 codes in total when subcodes are included. Three of the first-level codes constitute the sections of Chapter 6, which are mechanisms, barriers, and facilitators. The remainder of the first-level codes represent the company and interviewee-specific information as well as additional coded segments that are not directly relevant but still kept for further analysis. Subcodes constitute the specific type of mechanisms, barriers, and facilitators for knowledge sharing and coordination, and the results emerging from their analysis are demonstrated in Chapter 6.

5.5. Descriptive Study Data

This section gives an overview of the data collected during the interview study by describing the interviewed experts, their experiences, and the large-scale agile organizations they have been part of throughout their careers.

5.5.1. Companies

Table 5.1 shows the organizations covered during the interview studies, with their anonymized names, sizes, industries, and the number of samples collected from each of them. Since the experts were encouraged to share their experiences from all large-scale agile organizations they have been a part of, the total number of samples depicted in Table 5.1 is larger than the number of interviewees who participated in the study. Even if the experts took part in more organizations in their careers, only the ones that the interviewees shared their experiences about were added.

5.5.2. Interview Partners

As outlined in Chapter 1, the interview participants are selected in a way that allows each expert to contribute to the study by providing their experiences from different points of view. Ghobadi [60] demonstrates that the perception of the barriers to effective knowledge sharing in agile software teams highly depends on the person's role in the organization [60]. Therefore, in order to prevent biased results that only represent the view of a limited number of groups, it was crucial to include a wide range of roles and experts with different backgrounds in the interview study. Table 5.2 summarizes each interviewee's experiences, including the roles they have taken in their careers in large-scale agile contexts.

During the interviews, each expert was encouraged to share their experiences throughout their careers, without restricting the scope to their current organizations. The main reason for this was to ease the process of gaining a broad overview of the subject. As described in Chapter 1 as one of the main motivations of this thesis, the literature review revealed that

Anonymized Name	**Size**	**Industry**	**Number of Samples**
ConsultCo1	195,000	IT & Business Consultancy	1
SoftwareCo1	112,000	Software Products	1
TelecomCo	10,000	Telecommunication	1
FoodCo	5,000	Food Services	1
InsurCo	158,000	Insurance	2
ConsultCo2	357,000	IT & Business Consultancy	3
TransportCo	322,000	Transportation	2
EnergyCo1	5,400	Renewable Energy	1
ConsultCo3	10,000	IT & Business Consultancy	1
SoftwareCo2	1,000	Autonomous Driving	1
SoftwareCo3	250	Retail Solutions	1
EnergyCo2	2,000	IT & Renewable Energy	1

Table 5.1.: The list of companies covered during the interview series

a comprehensive study aiming to answer the research questions outlined is lacking in the existing literature. Therefore, shifting the focus from individual organizations to the overall experiences of the experts was an attempt to broaden the study's perspective, which was observed to work quite well as the interview partners were highly eager to talk about their past experiences in an informative and elaborate way. In addition, this approach also enabled a natural exchange between the interviewees and interviewers which, consequently, facilitated a better interview experience.

In total, 11 experts reported having experience in 16 distinct roles in large-scale agile contexts throughout their careers, as shown in Table 5.2. Compared to the number of roles that would have been collected if we restricted the interviews only to the current organizations (8), this means a twofold improvement in data collection and, as a result, an effective way to mitigate some of the limitations outlined in Chapter 7 regarding the data collection procedure.

Table 5.2 also shows the experience levels of each expert, both agile-specific and overall separately. For the experts who reported their experience levels as rough estimates rather than providing an exact number of years, the corresponding entries are represented by intervals. This was mainly because some interviewees had certain gaps regarding agile in their careers, and adding up the separate experiences was not immediately clear to them. However, the exact years of experience are not critical for this study, and thus, they are not pursued further, as rough estimates are sufficient to judge the experience level of an expert.

ID	Roles	Companies	Experience in Large-Scale Agile	Experience Overall
E1	Software Developer Technical Project Lead Development Lead Solution Train Engineer Quality Assurance Manager	ConsultCo1	20 years	30+ years
E2	(Domain) Scrum Master	SoftwareCo1	10+ years	10+ years
E3	Software Developer Project Manager	TelecomCo FoodCo	4 years	4 years
E4	Product Owner	InsurCo ConsultCo2	5 years	5 years
E5	Chapter Lead	InsurCo	3 years	10 years
E6	Lead Architect Scrum Master	ConsultCo2	8 years	8 years
E7	Business Analyst Requirements Engineer Rollout Manager	TransportCo	4 years	4 years
E8	Scrum Master Product Owner	TransportCo	9 years	35+ years
E9	Change Manager Product Owner	ConsultCo2 EnergyCo	1 year	10 years
E10	Machine Learning Engineer	ConsultCo3	2 years	5 years
E11	Scrum Master Product Owner Development Lead Software Architect	SoftwareCo2 SoftwareCo3 EnergyCo2	20+ years	20+ years

Table 5.2.: The list of interviewed experts and their experiences

6. Results

This chapter presents the results of this thesis by combining the findings from the interview study and the systematic literature review. Section 6.1 presents the knowledge sharing and coordination mechanisms identified. Section 6.2 describes the barriers, and Section 6.3 describes the facilitators in large-scale agile environments regarding effective knowledge sharing and coordination by discussing the benefits, trade-offs, and application contexts.

6.1. Knowledge Sharing and Coordination Mechanisms

Our interview study identifies 41 knowledge sharing and coordination mechanisms that are used in large-scale agile organizations. Similar to the classification of coordination methods proposed by Berntzen et al. [59], we group the identified mechanisms under three categories: meetings (30), tools (7), and organizational structures (4). Even though the number of online tools is 15 in total, only the categories they belong to are considered knowledge sharing and coordination mechanisms, as described in Section 6.1.3. The following sections describe each category and provide comprehensive lists of the mechanisms identified by the interview study. Although each mechanism might have its own initial goals of supporting knowledge sharing or coordination, our results suggest that, in practice, these two practices often occur simultaneously via the mechanisms identified, with certain exceptions. Therefore, all meetings, organizational structures, and tools are presented jointly, without differentiating them purely as a knowledge sharing or coordination mechanism. Instead, the primary focuses of each mechanism are outlined in the respective tables in each section.

6.1.1. Meetings for Knowledge Sharing and Coordination

Table 6.1 outlines the extensive list of meetings reported by the experts for knowledge sharing and coordination in large agile organizations. The list includes meetings that are commonly used by many organizations as prescribed by specific frameworks, such as the Daily Scrum [25], as well as the company-specific ones that are introduced by considering the needs of the organization in question, such as the meetings combining certain roles as in InsurCo (E5). While the information regarding the fundamentals of the framework-specific meetings can be found in Chapter 2, some additional contexts about the company-specific meetings are described in this chapter.

Our systematic literature review findings align with the interview study's outcome regarding the meetings for knowledge sharing and coordination in large-scale agile contexts. The results show that the meetings prescribed by the most popular agile frameworks are actively used

for knowledge sharing and coordination purposes in many organizations. The identified meetings resulting from the literature review include but are not limited to the following, with some of the publications reporting their usage: Stand-up meetings [65] [66], lunch talks [27] [67], seminars [27] [65] [67], retrospectives [27] [65], demos [27] [65], Scrum-of-Scrums meetings [14] [27] [65] [67], workshops [65], architecture meetings [67], PO Alignment [67], open space [27] [67], forums [27] [67], coffee breaks [67].

For each meeting outlined in Table 6.1, the main purpose of the meeting is marked in the respective column. However, it is important to note that most of the meetings identified are observed to include both knowledge sharing and coordination efforts either directly or indirectly. The following subheadings give some examples of such occasions. Therefore, this thesis does not classify the identified meetings between knowledge sharing and coordination and rather assesses their contribution to both of them with equal diligence.

When asked explicitly, the experts did not differentiate between the meetings regarding their benefits and proposed that all have equal importance for successful knowledge sharing and coordination in large-scale agile settings. However, the context in which they are useful differs, and the following subheadings present the definitions and the usage contexts of some meeting types listed in Table 6.1.

Meetings for Certain Roles

As described by E5, InsurCo introduces a certain type of meeting specifically designed for the Product Owners, Chapter Leads, and Agile Masters. It is a meeting that includes all the members of these roles in the organization, the main goal of which is to bring the whole large-scale agile structure to life (E5). The Agile Master role is introduced to encapsulate all agile teams that are not necessarily working in the Scrum framework, but the main idea of the role is similar to the role of Scrum Master (E5). The meetings are moderated with the help of external colleagues and are mainly used for coordination purposes (E5).

Agile Coffee Break

Agile Coffee Break is a weekly one-hour meeting for knowledge sharing at ConsultCo2 (E6). It has a specific agenda, and the participants are encouraged to talk about what their teams are currently doing (E6). It is particularly useful for having a common understanding as a large-scale organization because even though all the teams in the large scope work towards the same goals as one big team, each team still has its specific applications and projects, and they are not always fully aware of what the other teams are doing (E6). Agile Coffee Break facilitates intra-team knowledge sharing by bringing members from different teams together and encouraging them to share their current knowledge with their colleagues (E6). By actively using these meetings, the main purpose of which is enabling intra-team knowledge exchange, obtaining a certain degree of awareness about the tasks of the other teams can help mitigate

the dependency awareness challenges outlined in the literature [17] [68], and consequently facilitate intra-team coordination indirectly.

Future Workplace

Future Workplace is a monthly meeting in which the participants plan ahead and share ideas about their roadmaps (E5). It is open to everyone, including the customers (E5). In this way, customers worldwide are able to know what the organization is working on and provide valuable feedback (E5). The representatives from every country and business unit are also able to ask for help regarding their specific needs (E5). The meeting is a great way of sharing knowledge and coordinating within the organization (E5).

Developer Conference

Developer Conference is a knowledge sharing event that takes place every year at SoftwareCo1, as described by E2. In SoftwareCo1, it is mainly used to get people together on a voluntary basis and facilitate knowledge sharing across the company (E2). The event has one global and several regional versions (E2). The global one takes place online and usually lasts two or three days (E2). The regional ones also often take place online, but they still have a more in-person nature (E2). During the events, every participant is able to choose from a variety of sessions that cover different topics to join (E2). These sessions are not focused on a specific product but rather on broader topics such as security and technology (E2). Guest speakers are also welcome to contribute to the event (E2).

Town Hall Meeting

Town Hall Meeting is a knowledge sharing practice that usually takes place every quarter (E9). It is organized as an open space and does not have a certain agenda (E9). It shows similar characteristics with the open space environments reported in the literature [27] [67], as well as the all-staff meetings identified during the interview study (E1, E7).

6.1.2. Organizational Structures for Knowledge Sharing and Coordination

The analysis of the interview study reveals four organizational structures for knowledge sharing and coordination practices in large agile organizations. Table 6.2 outlines the identified organizational structures with their primary purposes. The list includes widely-known constructs, such as Guilds implemented at TransportCo, as well as company-specific implementations, such as the story-specific coordination role introduced at ConsultCo2 (E6). This section presents three of the identified constructs under the following subheadings. Guilds are briefly described in Chapter 2.

Meeting	Knowledge Sharing	Coordination
Sprint Planning		✓
Sprint Retrospective	✓	
Sprint Review	✓	
Sprint Demo	✓	
Daily Scrum		✓
Scrum of Scrums Daily		✓
PI Planning		✓
PI Demo	✓	
PI Retrospective	✓	
Backlog Refinement		✓
PO Alignment		✓
Meetings for Certain Roles		✓
Future Workplace	✓	
Kick-Off		✓
Release Kick-Off		✓
Scrum Master Sync	✓	
Technical Internal Meetings		✓
Level-Specific Meetings	✓	
Internal Coordination Meetings		✓
Lessons Learned	✓	
Intra-Team Knowledge Sharing Session	✓	
Internal Knowledge Sharing Session	✓	
Lunch Talks	✓	
Agile Coffee Break	✓	
All-Staff Meetings	✓	
Town Hall Meetings	✓	
Developer Conference	✓	
Trainings	✓	
Hackathons	✓	
Forums	✓	

Table 6.1.: Identified meetings in the interview study

Story-Specific Coordination Role

ConsultCo2 introduces a coordination role that ensures everything regarding a story is successfully maintained and completed (E6). It is not a separate, independent role in the classical sense but is taken by the existing team members, specifically developers (E6). It is classified under the organizational structures, as it extracts a set of people from the current team members, assigns them a certain role, and in this way, forms a new squad of people having additional responsibilities besides their regular ones. The experts having this role are responsible for creating all necessary subtasks and allocating the right tasks to the right people (E6). By the end, this role ensures that the final story is fully functional, all the requirements are met, and deployed on the integration environment before being presented to the customer (E6). The expert from ConsultCo2 where this construct was introduced as a coordination mechanism talks about a specific use case as follows:

> *For example, if you have nothing to do, you can then go to (this role) and say, "Hey, your story is the highest one in the sprint scope. Do you need someone to work on it? Or is everything else blocked?" And then (they) can say, "OK, so first two, (or) three subtasks can be taken on, or everything is blocked, please go and find something else to do."*

This role aims to fill the following coordination gaps arising in large-scale agile development environments regarding the management of tasks:

- No surveillance on the integrated individual feature branches.
- Unawareness of what to do with the open tasks.
- Unawareness of which additional team members could be allocated to the tasks to finish them earlier.

Providing surveillance on the broad overview of the integrated branches is critical to ensure the integrated solution works as expected at the end (E6). But when nobody is particularly allocated to take care of this problem, often nobody checks the resulting feature, or even if somebody does, it is too late (E6). Coordinating the allocation of tasks to the appropriate team members by a specific role is essential because, without a specific role, people often do not take on these coordination efforts themselves during the dailies (E6).

Communities of Practice (COPs)

Communities of Practice are observed to provide certain benefits regarding the efficient usage of the knowledge that team members have [69]. The results of our interview study show that they are widely used for knowledge sharing purposes in large-scale agile environments (E1, E2, E7, E11). In COPs, people talk about various topics such as coding guidelines, new tools, new frameworks, and best practices (E1). They exchange ideas and experiences, and learn from each other while potentially teaching their colleagues certain topics that they are knowledgeable about (E2). Some organizations implement multiple COPs that focus

on different topics, such as frontend development, backend development, architecture, and security related COPs at TransportCo (E7). In this setup, one member from each team whose expertise suits one of the COPs joins the respective COP (E7).

Although the main objective of COPs is to share knowledge, they also serve as an indirect coordination mechanism in certain contexts (E7). For instance, E7 reports that the participation of one team member from all teams in COPs enables teams to coordinate the dependencies and the topics that affect all teams (E7).

Despite their wide usage as a knowledge sharing mechanism, certain measures need to be taken into account so that their benefits can be reaped to their full potential. It is highly important to lead and guide COPs properly in order to be able to put the knowledge generated in COPs into action (E11).

Mentoring Programs

A mentoring program is an organizational construct working as a knowledge transfer mechanism between the generation of employees (E11). Contrary to the common belief, these structures do not just help inexperienced professionals, but also enable experienced experts to learn from their younger colleagues (E11). In these programs, seniors learn how to explain and teach topics better, while juniors utilize the knowledge of seniors (E11). Overall, the expert considers it a successful knowledge sharing mechanism (E11).

Organizational Structure	Knowledge Sharing	Coordination
Story-Specific Coordination Roles		✓
Communities of Practice (COP)	✓	
Mentoring Programs	✓	
Guilds		✓

Table 6.2.: Identified organizational structures in the interview study

6.1.3. Tools for Knowledge Sharing and Coordination

Our interview analysis reveals 15 tools that are actively used for knowledge sharing and coordination purposes in large-scale agile organizations. These tools are grouped across seven main categories according to their primary purposes and outlined in Table 6.3, together with their contribution to knowledge sharing and coordination practices. Our results suggest that even the tools that are not primarily designed for knowledge sharing or coordination

purposes, such as design and requirements management tools, can still serve as mechanisms for knowledge sharing and coordination, depending on the way they are used. This aligns with our findings regarding the effective usage of tools as a facilitating factor for knowledge sharing and coordination, which is demonstrated in Section 6.2, and once again highlights the importance of their exact implementation. As the functionalities of the tools are defined not solely by the product itself but also by how they are exactly used, the main focus should be placed on the usage details instead of their rollout in the organizations in order to understand their contribution to knowledge sharing and coordination activities.

Category	Tools	Knowledge Sharing	Coordination
Communication	Microsoft Teams	✓	✓
	Slack		
	Outlook		
	Zoom		
Documentation	Jira	✓	✓
	Confluence		
	SharePoint		
Version Control and Repository	Git	✓	✓
	GitHub		
	Bitbucket		
	Nexus		
Requirements Management	Doors	✓	✓
Design	Figma	✓	
Visual Workspace	Miro		✓
Online Whiteboard	ConceptBoard	✓	✓

Table 6.3.: Identified tools in the interview study

6.2. Barriers to Effective Knowledge Sharing and Coordination

The results of this thesis reveal a variety of barriers to effective knowledge sharing and coordination in large-scale agile environments. In this section, identified barriers are presented and put into perspective by referring to the findings of the literature review whenever appropriate.

6.2.1. Human Factors

The results of our interview analysis suggest that several human factors affect the efficiency of knowledge sharing and coordination practices in large-scale agile setups. This section presents these factors under the following subcategories.

Personality Traits

In large-scale organizations where a great number of people work together, it is natural to expect that not all employees have similar personality traits. For instance, while some people tend to be more open to talking about any topic, others might be shy about speaking up (E6). Our interview study reveals personality differences that deteriorate the efficient implementation of knowledge sharing and coordination practices in terms of the following personality aspects:

- Detail-orientedness (E1)
- Extrovertedness and openness to talking (E2, E4, E6)
- Need for proving oneself (E5, E10)

Detail-orientedness refers to how much a person is interested in fine-tuned information on a certain topic. In our interview study, this personality trait is observed in the context of implementing the Scaled Agile Framework (SAFe) in large-scale agile organizations (E1). E1 argues that the previous versions of SAFe were not as fine-tuned as the current version, and there was more freedom to define company-specific functions, but the current version defines everything in more detail (E1). In addition, a similar dissimilarity is observed among different people, as some of them like everything to be close to the books and clearly defined, while others prefer to define things themselves. The interviewee highlights that this dissimilarity can hinder the effectiveness of coordination, as it results in difficulties in terms of finding the right balance to efficiently coordinate the SAFe implementation (E1).

E4 argues that extroverted colleagues are more open to sharing their knowledge, and introverted associates sometimes underestimate the importance of knowledge sharing (E4). This shows up as a challenge, particularly because the introverted team members tend to take on their tasks in the sprint planning and implement them during the sprint without thinking about sharing anything regarding the progress, such as the faced problems and their solutions to them, which might possibly be useful for other colleagues (E4).

Related to the extrovertedness, analysis of the interviews also suggests that openness to talking is an important personality trait that facilitates knowledge sharing and coordination (E6). People usually do not hesitate to provide valuable ideas when asked directly, but not all people do so without being asked explicitly (E6). People who are not open to talking by nature usually stay silent in situations where something needs to be decided, but being bolder and speaking up to express agreement or disagreement about certain topics is something that is highly appreciated by the meeting organizers and considered a facilitator for effective knowledge sharing and coordination (E6).

E10 reports that people who are in a state of mind to prove themselves create a barrier to effective knowledge sharing and coordination (E10). A similar tendency is also reported by E5 as follows: *People behave differently because they want to show off and show to the chapter lead, "OK, I am doing a good job, [...] I am important here."* In these situations, the main reason for the emerging inefficiency comes from the fact that these people have a tendency to set up redundant meetings that do not serve any purpose regarding the actual tasks of the organization (E10). In these meetings, they like repeating the important points and do not create real value (E10). When people with this character join, especially a team of software development practices, knowledge sharing and coordination activities are hindered substantially (E10).

Language

Different native languages among the members of large-scale organizations are natural in today's globalized business world. In our interview study, several experts highlighted that this can hinder effective knowledge sharing and coordination in certain situations (E1, E3, E9). When an organization operates in multiple countries with different native languages, language changes in between result in a certain loss in communication (E1). Even if all the members speak English as the accepted common language in the organization, it still does not solve the entire problem as it is much harder to express oneself in a non-native language (E3). This also causes some teams to switch to their native language internally, but this only increases the language gap even further (E3).

Cultural Differences

Similar to the languages, cultural differences are also natural in large-scale organizations nowadays. Our interview study reveals that this can hinder knowledge sharing and coordination on some occasions (E1). E1 claims that cultural differences, just like language differences, result in a certain loss in communication (E1) and gives a concrete example as follows:

> *The customer(s) I am working (with) at the moment, (since) the product does not exist in India or Vietnam, do not understand why (we are doing this).*

Lack of Experience

Our interview study reveals that lack of experience has a hindering effect on effective knowledge sharing and coordination (E6). E6 elaborates on this as follows:

> *A lot of our work is importantly based on feelings. So you have your gut feeling, and with your gut feeling, you are usually right. [...] People from the outside who have done the same kind of dialogue several times have a good understanding (of) how complex it is to develop this dialogue, but if they have always done this with one framework and now they have to work with a different one, that might be tricky and working different way, then this is a challenge. Then you need the people who have worked in this special case, and they know, "OK, when I have to change this, I also need to generate this, I have to run this pipeline, I have to do this and this and this." [...] Then it is much easier than someone who has no idea of the application.*

Besides the above-mentioned human-related factors, our interview study also identified the following related barriers: Resistance to change (E1, E9) [70], blame culture (E3), and lack of trust (E3) [71]. However, since the contexts in which they appear are not elaborated further by the experts, they are not presented in separate subheadings.

6.2.2. Remote Work

Remote work has become increasingly popular in recent years [72]. Especially after the pandemic, many organizations started to adopt remote work as a commonly used work style. As a result, virtual teams are becoming a crucial focus point for knowledge sharing [44] and coordination activities. However, despite its benefits and wide application, remote work brings its own set of challenges which tend to escalate with the existence of geographically dispersed team members even further [70]. Although the overall advantages and disadvantages of remote work are beyond the scope of this thesis, this section describes the barriers to effective knowledge sharing and coordination in large-scale environments resulting from the remote way of working. As an emerging facilitator for knowledge sharing and coordination discovered by this investigation, emotional bonding is discussed separately in Section 6.3.4.

To begin with, it is important to note that this study does not classify remote work as a suboptimal practice or as something that needs to be completely avoided, unlike the majority of other barriers identified in this thesis such as insufficient documentation and misinterpretation of agile values, which are covered in the upcoming sections. Instead, this study outlines the related challenges resulting from this particular style of working and aims to elaborate on their potential consequences.

Despite the challenges it brings, including remote team members has various advantages for the success of an organization in the medium and the long term [70]. With programs of very large sizes, remote work often becomes a necessity rather than an option, especially from the cost point of view (E1). Not surprisingly, all experts in our interview study reported that

they experienced working remotely at least at some point in their careers, most of them being after the pandemic. These remote work settings include both fully remote options and hybrid ones. In addition, all interviewees named at least one hindering factor to effective knowledge sharing and coordination that originates from remote work settings. Our interview study reveals the following obstacles resulting directly from working remotely.

Limitation of Ad-Hoc Exchanges

When people work from different locations, the possibility of spontaneous exchanges reduces substantially (E2, E3). The results of our systematic literature review point out that ad-hoc meetings are considered an integral part of knowledge sharing and coordination practices, and can even be considered more efficient than scheduled meetings based on the findings of several case studies [56] [67] [73]. Consequently, diminishing the occurrence of ad-hoc meetings acts as an obstacle to effective knowledge sharing and coordination in large agile organizations. The role of ad-hoc meetings as a facilitating factor for effective knowledge sharing and coordination is discussed in more detail in Section 6.3.6.

Inability to Use Physical Tools

Remote work settings, by nature, do not allow employees to use physical tools such as post-it notes and whiteboards. Even though their online counterparts are widely used, their usage is not considered as effective as the usage of the originals (E3). One expert (E3) explains why this is the case as follows, with a concrete example:

> *If we were in the same location, I would always suggest having some post-it notes, (which is) very traditional. Just write [...] what the task is (and) who is responsible. And you can move the post-it notes from one place to another. And when people forget about their tasks, they can just instantly go there and see what is next or what they need to do. However, in Microsoft Teams, I believe that my team currently only opens the tasks just before the meeting. And I always try to show them in the meeting (by) the screen share, the task board. But I don't think that they are actively doing anything between the meetings.*

In other words, the inability to use physical instruments in an office environment reduces the chances of the team members to walk by, for instance, the post-it notes that summarize the current tasks or any other information that might be relevant for them, and thus, the chances of obtaining useful knowledge in an aleatoric way. Similarly, using whiteboards to sketch and explain certain topics to colleagues working closely is considered an effective way of communication between multiple people, but remote settings also eliminate this opportunity (E4).

Inability to Overhear

Communication among a group of people does not always occur in a pre-defined fashion where one party initiates a conversation with clearly set goals and expectations in their minds.

Often, people find themselves listening to a conversation that happens to take place near them, without even intending to listen to it in the first place (E5). Especially in large-scale organizations with large headcounts, this can work as a spontaneous way of communication which happens by chance, as described by E5 as follows:

> *Sometimes I am not sure who is interested in (a) specific information, because I do not know that they are working on a similar topic for a different customer. And if I knew that, I would tell them, but I do not know this. And that is why I am just talking (with others). And then someone overhears it, "Oh, you are doing the same stuff, or you did something similar. [...] So let's align and avoid duplicate efforts." [...] So, information sharing with the right people at the right time... That is difficult when you're not in the office.*

As E5 outlines, overhearing also helps identify for whom the knowledge possessed by an employee can be useful in the organizations (E5). With the lack of such a facilitating mechanism, knowledge sharing practices are hindered (E5, E6). In addition, the inability to overhear resulting from working remotely also forces employees to schedule meetings that occupy larger time slots for very brief topics that could have been clarified in an office environment with a simple overhearing (E6). Consequently, knowledge sharing does not occur as effectively as it could have been, since the resulting situation is complicated to handle (E6).

Inability to Observe Colleagues

The opportunity to observe colleagues in their working environments and have some eye contact with them has certain benefits for effective knowledge sharing and coordination activities (E5, E6, E8). These benefits can be classified into two categories: one-to-one contexts, and meetings.

When large-scale agile organizations work co-located, team members can individually see whether their colleagues are working in deep concentration mode or a more relaxed fashion so that they can decide better whether they should approach them to ask a question or to share something at that specific moment, or they should rather wait for a more appropriate time (E8) [74]. This can particularly be useful when certain people are unwilling to disturb others [71]. In online settings where seeing the current state of colleagues is almost never possible, approaching a colleague for knowledge sharing or coordination purposes is a more challenging experience (E8). Even though the status indicators in online communication tools such as Microsoft Teams and Slack attempt to tackle this barrier, they are not always fully trustable, as the team members often do not regularly update their status information [74].

The same obstacle also diminishes the knowledge sharing and coordination efficiency in the context of meetings (E6, E8). When a presenter waits for some input from their audience, one of the smoothest ways to accomplish this is to look the respective people in the eye so that they can understand some input is expected from them (E6). This subtle mechanism is unlikely to occur in remote work settings as specific colleagues cannot be addressed in large

audiences just by looking at the computer screen (E6). The expert describes what happens as a result of this barrier as follows:

> *Everyone is just quiet, all mics are muted. You would say something, and no one really answers. [...] So, if you would have a virtual solution for this, it would be great.*

The expert further reports an initiative at their organization (SoftwareCo1) to tackle this issue by creating online rooms that aim to imitate real-life work environments so that colleagues can interact with each other and observe what the others are doing (E6). Yet, the experience is not similar to the co-located settings (E6).

In remote work settings, it is also difficult for the presenter to assess whether people are actually listening to the content of the meeting or doing some side work in the background (E8). In situations that require the attention and feedback of all participants, this barrier makes it significantly difficult to get the necessary information from the audience and hinders the knowledge sharing and coordination efforts consequently (E8).

Inability to Use Body Language and Facial Expressions

Working from different locations reduces the amount of body language and facial expressions that can be used and perceived during knowledge exchange and coordination practices (E5, E6, E8, E11). Although facial expressions are expected to be conveyed thanks to online video communication tools, some people are still reluctant to turn on their cameras (E5, E6), and sometimes even to participate in meetings (E5). Non-verbal communication is considered an important part of the interaction between colleagues and is highly restricted by online settings (E8). The implications arising from facial expressions and body language, even if the person does not speak at that moment, get lost in the conversations when people do not work in a co-located setting (E5).

Even though working remotely is not necessarily a bad practice and can have its own advantages, such as better concentration (E8), our results still suggest that several emerging factors resulting from remote work settings hinders efficient knowledge sharing and coordination practices in large-scale agile organizations. A hybrid approach can help maintain the advantages of both approaches, as experts remark that working remotely is also something they prefer at least from time to time (E4, E5, E7, E8, E9, E10).

6.2.3. Time Zone Differences

With the increasing opportunities to work from everywhere, thanks to the advent of online communication tools, organizations have started to employ people from different continents to enter new markets, lower costs, or achieve certain goals that are related to their business needs (E1). Nowadays, many companies have off-shore teams or customers to work together towards common goals. As a consequence, organizations become prone to suffering from time zone differences between their members. Even though some organizations try to solve

this by clearly defining the working hours for their off-shore colleagues in a way that they overlap as much as possible (E5), this requires the distant colleagues to work in the evening or even at late night depending on the location. Conversely, some organizations do not find this ideal and, thus, do not ask their distant colleagues to adjust their working hours according to them (E1). But then, the time zone issue remains a barrier that needs to be addressed (E1).

In our interview study, three experts (E1, E9, E11) reported time zone differences as a barrier they experience in front of effective knowledge sharing and coordination. When people in various time zones need to work together, certain disruptions arise in the workflow as people spend too much time updating themselves about what the other colleagues ahead in time had done, until they started working that day (E11). This further leads to an isolated way of working where people become reluctant to take on responsibilities (E11).

While one or two hours of difference is not considered a problem (E2, E3), a six-hour gap can substantially threaten the effectiveness of knowledge sharing and coordination practices (E1).

6.2.4. Insufficient Documentation Efforts

As outlined in Table 2.1 as one of the values suggested by the Agile Manifesto, Value 2 states that working software is preferred over comprehensive documentation efforts [1]. However, our results show that this value might sometimes be misinterpreted by the members of agile teams, as described more in detail in Section 6.2.7.

Even though it is unclear whether all the insufficient documentation efforts stem from the misunderstanding mentioned above, lack of proper documentation remains a challenge regarding effective knowledge sharing and coordination in large-scale agile contexts. In total, four experts in our study emphasized the insufficient documentation efforts as a factor that diminishes the knowledge sharing and coordination in their organizations (E4, E6, E7, E10).

Experts explain this barrier with examples from the specific use cases of certain online tools. When using online planning and documentation tools such as Jira[1], explicitly writing the current status of the committed code piece to the feature branch as a comment is considered a highly beneficial way of sharing knowledge and coordinating with the team members, as it enables other colleagues to understand and possibly work on the same feature in the event of unexpected absences (E6). However, this ideal state is usually not the case (E6). This problem escalates especially when people are highly focused on their tasks because, without sufficient documentation, it is not possible to keep everything short and prevent taking too much time (E6). In a similar manner, documentation in Confluence[2] requires additional efforts to keep it structured and up-to-date (E7, E10). However, these efforts are often not realized in practice (E7, E10). Mostly, people add information in Confluence, but they do not delete the ones that are not relevant anymore (E10). This creates an ever-growing knowledge base in which the

[1]https://www.atlassian.com/software/jira
[2]https://www.atlassian.com/software/confluence

information is outdated and harder to trust and, thus, very hard to keep track of (E6, E7, E10). In addition, having a Confluence structure that is inherited from the former ones also restricts the access of the new team members to the knowledge they need, as they are not familiar with the inherited structure (E7). As a result of these, employees often find themselves in a position where they are not familiar with some fundamental knowledge they are supposed to have (E7, E10).

Practicing routines that facilitate updating the relevant documentation as soon as any change happens, such as regularly adding documentation needs to the task lists, is considered a helpful strategy to overcome this barrier (E6). Further findings about using documentation as a significant facilitator for knowledge sharing and coordination are demonstrated in Section 6.3.1.

6.2.5. Involvement of External Parties

Organizations often integrate external employees to strengthen their workforce. Our results show that the inclusion of externals in large-scale agile organizations creates a complex knowledge sharing medium and leads to certain inefficiencies in knowledge sharing and coordination practices if not addressed properly. The main reason for these inefficiencies stems from the specific legal regulations that prevent people from sharing their knowledge with external colleagues (E5). Even though externals work closely with the permanent team members towards a common goal, they do not have access to certain information, such as who is going to be on holiday and who is on sick leave, due to the regulations of the workers' council (E5). Therefore, creating some private channels is necessary for online communication tools to share information that is not relevant to the externals, but those private channels are observed to be highly inefficient due to their limited functionalities (E5). One example is the inability to use some available add-ins for Microsoft Teams channels in newly created private channels (E5). Secondly, these private channels are reported to act more like separate SharePoint[3] pages that are difficult to share data with. As a further attempt to tackle these problems, creating multiple teams for each agile team and another one including all members with a separate private channel is observed to worsen the existing situation regarding knowledge sharing efficiency even more (E5). In the resulting setting, employees need to check different channels and different teams in their online communication tool regularly to ensure that they are up-to-date with the newest knowledge available to them, but this is often not easy to practice (E5). In addition, this setup also diminishes the efficacy of one of the facilitators for knowledge sharing, which is transparency (E5). Having different teams in the online communication tool for each agile team prevents team members from keeping track of what the other teams are doing and communicating over, and thus, reduces the level of transparency significantly (E5).

[3]https://www.microsoft.com/en-us/microsoft-365/sharepoint/collaboration

Similar efficiency issues arise when too many external providers are outsourced during the development process in large-scale agile organizations (E5). This is mainly because the external provider potentially works in a very different way than the organization (E1, E5, E6), probably not even in an agile way, but all the teams still need to work closely to accomplish the same set of goals (E5). In addition, some external providers might be reluctant to share their knowledge regarding a certain topic to keep the organization dependent on them and stay an important part of development (E5).

From the team spirit perspective, integrating the externals does not seem to be an issue (E7). However, addressing them formally when something urgent arises remains a coordination challenge (E7). Since the external parties have distinct autonomy, it is not always possible to push them to finish certain tasks at a certain point in time, even though it is believed to be absolutely necessary for the product and business goals (E7). Creating a team spirit is observed to work well in this sense. When the externals feel that they are part of the team, they do their tasks in a timely manner without the need to push them to do so explicitly (E7).

Although not all large-scale agile organizations necessarily interact with external employees or providers, it is very likely that they can experience such contexts at some point in today's business world. Therefore, knowledge sharing and coordination with external parties need to be addressed as a significant barrier in front of effective knowledge sharing and coordination, to be able to reap the benefits of relevant practices put into place.

6.2.6. Lack of Transparency

In our interview study, three experts mentioned lack of transparency as a barrier they observed in front of knowledge sharing and coordination in large-scale agile organizations (E5, E7, E11). E11 highlights the importance of making knowledge transparent so that people feel proud of what they have, which consequently facilitates knowledge sharing, and further expresses that large-scale agile frameworks such as SAFe do not sufficiently stress the significance of transparency (E11). E7 argues that their meeting results are not transparently communicated in their current organization, and this causes their meetings to become less efficient than they could be (E7). Similarly, E5 remarks on the inefficiencies arising from their Microsoft Teams setup that diminishes transparency, as described in Section 6.2.6, and thinks that transparency could have helped them to work together more effectively (E5).

6.2.7. Misinterpretation of Agile Values and Principles

In our interview study, certain values and principles proposed by the Agile Manifesto [1] are observed to be misinterpreted by some members of agile teams in large-scale organizations (E9). Our results reveal that Value 2 in the Agile Manifesto, as depicted in Table 2.1, often makes people underestimate the benefits of documentation efforts and start not documenting at all (E9). E9 talks about this misinterpretation as follows:

> *The key challenge, in my opinion, is working in an agile way is often misunderstood. So, some people think that working agile means not documenting at all, and I think it is crucial to document. But I heard people saying, "We work in an agile way." [...] They are just trying to justify that they were not organized, not structured, not really aligned as a team, and not really taking care of documentation. That often led to the project failing. So, I think it makes sense to document whatever you do, even if it changes quite often.*

Consequently, misinterpretation of agile values and principles shows up as a barrier in front of knowledge sharing and coordination, as proper documentation efforts are identified as a facilitating factor by many experts in our study (E1, E3, E4, E9, E10). This marks the clear communication of agile values and principles as an important step towards enabling effective knowledge sharing and coordination in large-scale agile environments. The facilitating nature of documentation efforts is further demonstrated in Section 6.3.1.

6.2.8. Unawareness

Our literature review reveals that unawareness shows up as a barrier to inter-team knowledge sharing and coordination in large agile organizations [17] [71] [75]. Our interview study further identifies two types of unawareness that hinder effective knowledge sharing in large agile organizations (E8, E11).

Unawareness of Other People's State

People often fail to assess the existing knowledge level of their colleagues and talk about the ideas in their minds without putting them into concrete contexts (E8). However, sharing knowledge does not always guarantee a complete understanding of the knowledge by the receivers, as the receivers often modify the input in their minds [58]. In addition, knowing who knows what helps with creating a shared mental model, which, consequently, facilitates better coordination [75]. Therefore, being aware of the receivers' state of mind and current knowledge level is critical to ensure that knowledge sharing occurs as intended and the teams coordinate efficiently. As demonstrated in Section 6.2.1, E1 highlights the significance of the receivers' current state by mentioning that their off-shore colleagues often do not fully grasp what they mean in certain contexts, since the product in question does not exist in their culture (E1). Similarly, E8 explains the importance of putting explanations into certain contexts with the following imaginary example:

> *Deep in mind, you do not even know that it is (the) knowledge other people do not have. [...] What would you expect about talking about the German word Mutter? It might be your mom, or it might be somebody who is really taking care of you. It might be a part of a screw where you put it together, the thing with a hole and screw. That is called Mutter, as well. So, if you use Mutter and you do not talk about what the context of that word (is), you lose people, and that is a big obstacle.*

Unawareness of One's Own State

Besides being aware of other colleagues' states to better communicate and share knowledge with them, self-awareness of one's own state is also observed to be an important factor affecting knowledge sharing in large agile organizations (E11). Unawareness of one's own knowledge is considered one of the most crucial barriers to effective knowledge sharing by E11 and is explained as follows:

> *People are not even aware of the importance of their knowledge. [...] (They think,) "I know about this, for me it is common sense, so why should I tell anyone that I know this? For me it is common sense, I would expect that many others know about this as well." So they're just not being triggered in the sense, but hey, you know about something that could be probably of worth for someone else!*

E11 further identifies two additional reasons why people do not share their knowledge. These reasons are demonstrated in Section 6.3.7 in the context of facilitating knowledge sharing.

6.2.9. Lack of Management Support

Our interview study exhibits that lack of management support is one of the barriers to effective knowledge sharing and coordination in large-scale agile organizations (E4, E9, E10). Managers often tend to prioritize the number of features to be rolled out over the quality of the overall product by focusing on the short-term goals (E4) and do not prefer agility as much as the teams (E3). However, they should rather emphasize the vision and the goals of the product or project in question (E10). When there is no explicit top-level management support, effective knowledge sharing and coordination practices become harder to implement in practice (E9).

6.2.10. Project Kick-Off

The project kick-off phase, just by itself, is reported to be a challenge regarding coordination by two experts in our interview study (E6, E10). E6 explains why it deteriorates coordination as follows:

> *It is always a challenge if you start a new project. Even (if) you have been doing Scrum for several years, and maybe even with more or less the same people in the same customer environment, every time you go into a new Scrum project, you have to understand how everyone estimates, how much the story point is worth, then what is an easy task, what is a complex task, all of these things. Even that is already (a) problem.*

Unlike the other barriers presented in this chapter, the project kick-off phase is, by its very nature, impossible to avoid completely. Although several attempts can be put into practice to make the onboarding a smoother process, it is usually thrown out of focus since it is a challenge experienced only by newcomers, and consequently, people tend to forget about

it once they pass this phase (E10). Yet, some organizations implement certain constructs specifically designed for the initial stages of the projects, such as the formulation of a small inception team [58]. In this construct, the team consisting of Project Managers, Technical Leads, and Business Analysts participates in the project inception and prepares documentation regarding the outcomes of the meeting [58]. The team assesses the project's viability, and if it kicks off, they share extensive knowledge about it with their teams [58]. This practice works as a mechanism for knowledge transfer [58], and we propose that similar constructs, possibly with different team formulations depending on the organizations, specifically designed for the project kick-offs can be highly useful to achieve effective knowledge sharing and coordination in large agile organizations, as they would address one of the barriers reported directly.

In addition, our interview study identifies the additional barriers to effective knowledge sharing and coordination, such as inadequate tools (E7), redundant meetings (E3), and lack of clear responsibilities (E7, E9), which are not covered in separate subheadings as they are not elaborated further by the experts. A comprehensive list of all barriers identified and their effect on knowledge sharing and coordination practices can be found in Table 6.4. In certain cases where the effect on the knowledge sharing or coordination practices is not clearly evident but still implied by the findings of the systematic literature review and the interview study, the term *indirectly* is added in the respective rows.

6.3. Facilitators for Effective Knowledge Sharing and Coordination

Our results highlight a variety of facilitating factors for effective knowledge sharing and coordination in large-scale agile environments. In this section, identified facilitators are presented and put into perspective by referring to the findings of the literature review whenever appropriate.

6.3.1. Documentation Efforts

Although Value 2 of the Agile Manifesto prioritizes working software over extensive documentation [1], the results of our interview study combined with our findings from the literature review suggest that thorough documentation efforts are still highly beneficial to improve effective knowledge sharing and coordination in large-scale agile organizations. In total, seven experts in our interview series revealed that a successfully implemented documentation strategy substantially helps with the different aspects of their knowledge sharing and coordination activities. E4 talks about the importance of documentation as follows:

> *People underestimate how important documentation is, so people say, "OK, we know it." but they always underestimate the fact that people leave (the company). If you have a good documentation, that helps in the future so much, not only because when you onboard someone and do the knowledge transfer easily, also in case you have an issue, [...] if you have a good documentation, you can solve (it) much better.*

Barrier	Knowledge Sharing	Coordination
Personality Traits	✓	✓
Language	✓	✓
Cultural Differences	✓	✓
Lack of Experience	✓	✓
Limitation of Ad-Hoc Exchanges	✓	✓
Inability to Use Physical Tools	✓	✓
Inability to Overhear	✓	indirectly
Inability to Observe Colleagues	✓	✓
Inability to Use Body Language and Facial Expressions	✓	✓
Time Zone Differences	✓	✓
Insufficient Documentation Efforts	✓	✓
Involvement of External Parties	✓	indirectly
Lack of Transparency	✓	✓
Misinterpretation of Agile	✓	indirectly
Unawareness	✓	indirectly
Lack of Management Support	✓	✓
Project Kick-Off		✓
Inadequate Tools	✓	✓
Redundant Meetings		✓
Lack of Clear Responsibilities		✓

Table 6.4.: Identified barriers in the interview study

The expert further explains that they built a habit of ensuring they do not close the sprints before the entire documentation is finalized, even if all the tasks in the current sprint are completed (E4). In this way, they enable easy access to the knowledge within the organization (E4).

The interview study identifies two main functions of documentation efforts while enabling knowledge sharing and coordination: Firstly, it serves as a bridge over which the knowledge sharing occurs between the generations of team members in an organization, and secondly, it replaces redundant meetings whenever possible.

Documentation as a Bridge Between Generations of Team Members

As people join and leave organizations, documentation is a crucial practice to maintain a link and ensure continuous knowledge transfer. Quit of the experienced team members, especially experienced developers in the context of software development, from the project or the organization results in the loss of important knowledge [76]. Three interviewees in our interview study particularly stated that extensive documentation helps reduce the gap between the different generations of team members in their organization (E1, E9, E10). E1 talks about the problems emerging with the lack of documentation as follows:

> *To be honest, this is a journey over time. When [...] you think five years in the future, then all the team members do not exist anymore because your business, as usual, is only five or ten percent of your initial set. (You make) efforts to set everything up, and then you still have no summary of the documentation. So here I am talking about communication over generations of people, let's say this way.*

Further, they explain how they attempt to cope with this issue in their teams, particularly from the perspective of enabling software developers:

> *For example, we have our rules that in the source code, you have to include with each change [...] the link back to the feature (in GitHub). So when in five years, someone is looking at the application and asking himself, why this is working in this way and not the other way, he always has the reference back to the initial feature. And this is, for example, something which is often forgotten.*

Similarly, E9 discusses the same issue from a broader perspective:

> *Because if, for example, someone is leaving the team and someone is joining, it makes it easier to have it all in one place, have it all documented like, "OK, that's where we were coming from. That is where the challenges lay." Also, if you think ahead of the retros and the lessons learned, it makes sense to document everything. Because then if you look back, you can find out, "OK, we will figure out what could have been done better or not."*

Supporting the views of E1 and E9, E10 puts emphasis on the onboarding process and argues that new members of an organization should ideally be able to find the knowledge they need to onboard themselves in the organization smoothly to be able to complete the whole process efficiently (E10).

Documentation as a Replacement for Redundant Meetings

One of the interviewed experts highlights the importance of documentation as a mechanism to eliminate redundant meetings, and consequently, to facilitate knowledge sharing and coordination practices as follows (E3):

> *If there is a platform (where) they can update as the changes occur, like they can change the status and everyone can go and look at that, how it is going. So it is not like, no, there are 10 tasks and they are going task by task. How is the first one, the second one? So they can update everyone doing the first one, they can update the status and you do not need to waste two hours every week.*

Considering the fact that time management is often an important aspect of knowledge sharing and coordination activities (E2, E3, E6), the elimination of certain meetings can help enable efficient knowledge sharing and coordination by increasing people's availability (E3).

6.3.2. Efficient Usage of Tools

As outlined in Section 6.1.3, our research identifies various online tools that are used for knowledge sharing and coordination directly or indirectly, such as Confluence[4], Jira[5], and Microsoft Teams[6]. However, the results of the interview study suggest how exactly these tools are configured and used in an organization highly impacts whether they facilitate knowledge sharing and coordination. When the online tools are not set up efficiently, they can become overwhelming for the team members [73].

For the software developers, E1 mentions the importance of a subtle but effective detail regarding the usage of the version control and repository tools, Git[7] and GitHub[8]. The interviewee argues that linking back to the feature with each change committed allows future developers to understand the implementation process much better, and thus facilitates knowledge sharing and coordination between the generations of developers in the organization (E1). Similarly, E4 highlights the importance of linking Jira tickets whenever a team member documents something, regardless of what is being documented on which platform by which individuals (E4). The expert emphasizes that an efficient setup of communication tools is the key to effective knowledge sharing and coordination (E4), and explains their approach to this as follows:

> *I think an efficient Microsoft Teams or Slack, whatever you are using, setup is crucial. [...] Sometimes people make the mistake that [...] they put, for example, business stakeholders all the time in a chat with the development team, with a huge development team. I find this*

[4]https://www.atlassian.com/software/confluence
[5]https://www.atlassian.com/software/jira
[6]https://www.microsoft.com/en-us/microsoft-teams/group-chat-software
[7]https://git-scm.com/
[8]https://github.com/

> *super inefficient. [...] We were using an efficient (Microsoft) Teams setup, [...] so we had a Teams group, and then multiple channels, [...] for example, (for) operations, for discussing issues from live systems, for development teams, (for) pure business topics, and so on. There were also secret ones, and only people who should really be notified were getting those alerts. So in case we had something for the development team, or for the release, or suddenly if we needed to contact someone, then yes, we shared the information always with everyone. [...] We do not need to put the developers in a chat with 50 people and disturb (them) all the time, so I think that is also very important for knowledge exchange: efficient usage of those chat tools.*

Several other experts also mention a similar, function-based setup of their communication channels (E6, E7, E9). E6 explains that they face a certain challenge while communicating via Microsoft Teams because of the different notification settings for channels and chats; the former notifies everyone, and the latter does not. In their organization, they avoid this by applying a Microsoft Teams configuration in which the channels are organized according to the interests of different subgroups, similar to the setup mentioned by E4. However, E6 additionally emphasizes the importance of having an overarching separate chat to inform everyone, and says:

> *"If it is something for everyone, then it is the main chat. For example, if I want to take a few days off, then I write this in the chat to see if everyone is alright with it. Everyone says and sums up, and so it is acknowledged by enough people (that) you can take off basically."*

E7 talks about the same issue but for another tool, Confluence[9], and considers its current usage at their organization as an obstacle for effective knowledge sharing and coordination (E7), implying the facilitating nature of an efficient setup:

> *Another obstacle [...] is our documentation in Confluence. This is huge, historically grown, and not very structured. So it is difficult to find the information if you want to find something. And basically, everyone uses their own bookmarks, (and) structure of bookmarks to access our Confluence pages. But it's not using the Confluence page hierarchy, because it is a mess.*

In line with the elaborations provided by E4, E9 expresses their view on the distraction arising from inefficient setup of communication tools with crowded chats, and further makes suggestions about how the ideal setup should be decided:

> *If you have [...] big chats, especially if you have like a large team of 15 people, and everyone is typing down [...] whatever they think about, it takes a lot of time, you get distracted, you read through it, and you are like, "OK, it does not really matter, it does not really affect me." But you are (still) distracted all the time. [...] (Therefore) I am not a big fan of having these big chats, but you should have like a really structured MS team (in Microsoft Teams).*

[9]https://www.atlassian.com/software/confluence

> *So, I think the team, once you start the project, should think about how to structure it in the best way possible, and that everyone is happy with and everyone thinks it is the best way to collaborate with each other.*

The results of our study suggest that it is wise to focus on the exact implementation and the usage of the tools rather than solely counting on their rollout in the organization if the knowledge sharing and coordination practices are targeted to become as efficient as possible. Especially for online communication tools such as Microsoft Teams and Slack, careful selection of participants for each chat, channel, or group is essential to make sure the knowledge that is intended to be shared reaches the relevant audience and does not disturb the other members that are not likely to be interested in the knowledge in question. In light of these findings, the next subsection extends the participant selection concept beyond the online communication tools and discusses its importance for the organizational structures and the meetings in general.

6.3.3. Proper Selection of Participants for Organizational Structures and Meetings

In total, five experts in our interview study explicitly reported the significance of selecting the right audience for a meeting or an organizational structure to facilitate effective knowledge sharing and coordination (E1, E2, E3, E7, E8). E1 argues that meetings are efficient only when the right participants are involved (E1), and, to make it concrete, gives an example of an inefficient setup where an all-staff meeting with 600 people takes place and a very specific topic that is only relevant for a small number of participants such as the length of a name field is discussed. In this case, the expert suggests that either the participants or the agenda of the meeting need to be revised in order to enable efficient coordination (E1). E2 puts emphasis on the time issues that might potentially arise if the team members are expected to participate in many different types of meetings without careful elimination because they would not have enough time to work on their actual tasks (E2). In a similar manner, E3 believes that not everyone needs to be involved in every meeting, and explains that in their current organization (FoodCo), they are putting special efforts to minimize the length of some of their Scrum meetings to keep them under 20 minutes so that the schedules of the participants can be relieved as much as possible. E7 considers including too many people in the meetings as one of the most important reasons why the meetings in their current organization (TransportCo) are not as efficient as they could be. The following quote from E8 summarizes the above-mentioned points as follows:

> *You really have to look at who should be part of the meeting, who is learning in the meeting, and who is bringing content to the meeting. So that is where you really have to look at, especially in a scaled structure where you have to have more meetings because every scaled agile structure brings with it more meetings and more overhead.*

However, the selection of the participants might not always be a straightforward task due to the trade-offs arising between letting the team members focus on their tasks and keeping them

up-to-date with regular meetings (E2). E2 explains his view on this and gives an example from his previous organizations:

> *In one of my previous roles, I had these situations that people were complaining that there were too many meetings. Then I said, "OK, then pick only the ones that are relevant to you." And some people picked half of them, some people picked most of them, some people picked none of them. And then they started complaining that they did not know anything. So there is no silver bullet here. It is like a trade-off that you need to find the balance. And also, once you find the balance, at this point in time it may work. The balance may be right one month from now, but two months from now or three, the situation may change and you may need to attend more meetings or fewer meetings, because the situation in the project or product changed. So even if you find the right balance, then there will be situations that you will need to adjust because of the situation around you.*

The results show that participant selection for the meetings and the organizational structures is a crucial facilitator for effective knowledge sharing and coordination, but at the same time, it is hard to achieve due to the complex nature of large-scale agile organizations, changing environments, number of people, and consequently arising trade-offs.

6.3.4. Emotional Bonding

In recent years, many organizations have adopted remote work at an increasing pace [72]. As a result, colleagues who would normally see each other occasionally in the office, during lunch, or at a specific event have substantially decreased their chances of interacting with each other in person. Although the overall advantages and disadvantages of remote work are beyond the scope of this thesis, we investigated the subject from the perspective of effective knowledge sharing and coordination and asked all interview partners whether they observe any difference between working remotely and being co-located with their colleagues in this sense. This subsection introduces one of the resulting factors that emerges from the interview study as a facilitator, which we describe as emotional bonding. The remaining aspects of remote work regarding knowledge sharing and coordination are discussed in Section 6.2.2.

In our interview study, eight participants highlighted the importance of having some type of emotional connection with their colleagues as a significant facilitator for effective knowledge sharing and coordination in their organizations (E1, E5, E6, E7, E8, E9, E10, E11). E1 and E5 remark that it is significantly easier to communicate with their colleagues when they can freely catch their facial expressions and look them in the eye so that they can receive the emotional elements of the communication (E1, E5). While talking about an instant language translation tool their organization (ConsultCo1) is about to roll out, E1 mentions the significance of the emotional part of communication as follows:

> *But it is not only the language translation. It is also when you listen to someone in their own language or their own saying. Then you understand, even in a nonverbal (sense), what is important for him or not. And this is [...] (the) emotional level of communication.*

Similarly, E5 talks about how they benefit from being able to catch the emotional elements in a conversation:

> *If you look into each other's eyes, especially during the retrospectives, et cetera, you can read more than you can ever via a webcam or a (Microsoft) Teams meeting or something. [...] Even if someone is not saying something, they are saying something. And if you cannot see that, that gets lost.*

E6, E7, E8, and E9 focus on the importance of getting to know their colleagues better in order to communicate effectively with them at later stages. E6 and E9 highlight that establishing a personal connection is particularly necessary in the initial phases of team formulation and kick-off to establish some bond, and E6 and E10 further emphasize that social activities such as going out in the evening, doing some sports, and coming together to drink some coffee help a lot to achieve that. E7 says:

> *Sitting together helps me to get to know my colleagues better and know how I could talk to them and somehow get emotionally on the same level, get to know each other. And this makes communication from remote then again easier.*

Emotional bonding helps facilitate effective knowledge sharing and coordination also by eliminating some of the barriers in front of it, such as an initial diffidence that some team members might experience in environments where they do not feel emotionally connected and comfortable (E10). The following quote from E10 remarks that practicing some events to personally connect people is beneficial even if the team members are not working co-located on a regular basis:

> *When you meet people there (outside) and then do the work remotely, you actually feel like, "OK, I know the people, and I can communicate, and I can ask questions." But when you never saw people ever and just communicated through Microsoft Teams, when you first started out, that is a really stressful experience.*

E8 argues that the meetings that are conducted with familiar team members are much more efficient than the ones that involve other teams, which is in line with the findings of Kuusinen et al. [49], because of already having an understanding of their personality, their language, and what they might be implying with certain words or even looks:

> *I think the meetings within the team are the most efficient ones because you have the same language, you have the same understanding, and you know each other very well. So you just sometimes need a look or a word to get the whole thing clarified. And in a meeting with another team, you sometimes really have to find whether you use or not use the same topic for the same words or the same words for very different topics.*

Our results encourage the implementation of mechanisms that enable members of an organization to connect with each other personally and emotionally, in order to ensure effective

knowledge sharing and coordination. In certain contexts, even the regular meetings prescribed by agile frameworks, such as Daily Scrum, can help form an emotional bond and team spirit in agile organizations [77]. Furthermore, bonding with colleagues and establishing a team spirit decreases the need for formal arenas for effective coordination [75]. Considering the fact that lack of team spirit is a barrier to effective knowledge sharing [71], enhancing emotional bonding is critical for facilitating better knowledge sharing practices (E6, E7, E8, E9). This is particularly important for large-scale organizations mainly because of the size and, consequently, the inability to establish personal bonds in a regular working schedule with multiple teams. Although this is not always straightforward to implement especially in international settings where people work from different locations, some organizations are already attempting to promote it by integrating various approaches such as introducing an instant online translation tool to eliminate language differences between off-shore colleagues and enable better emotional connection (E1), and creating online rooms that aim to imitate real-life work environment to promote spontaneous interaction between colleagues (E6) [74].

6.3.5. Flat Hierarchy

During the interviews, we asked each interviewee whether they observed any relationship between various company structures and the effectiveness of knowledge sharing and coordination practices in their organizations. The results signal that avoiding strict hierarchical structures and reporting lines can serve as a facilitator particularly for effective knowledge sharing in large-scale agile organizations, as three experts explicitly stated that they prefer flat hierarchies to enable knowledge sharing (E4, E9, E10). One interview partner expressed that enabling effective knowledge sharing and coordination is not about organizing the structures of the organizations, but about organizing the topic itself (E11). The remainder of the interview partners expressed that they are either indifferent between these views, or that it depends on the exact context, but none of the experts strictly preferred classical structures and reporting lines over flat hierarchies. In environments where there is almost no hierarchy, it is observed that internal training sessions are much more effective, as experienced colleagues tend to share their experiences with their younger colleagues without restricting themselves to the pre-defined hierarchies (E4). In contrast, training sessions in more hierarchical organizations tend to have a more traditional focus on topics such as leadership essentials, and thus, knowledge sharing does not take place as effectively as it does in flat structures (E4). Since senior team members do not necessarily possess the best knowledge, and a junior may well have the best information about a certain topic, removing hierarchies as much as possible and encouraging people to share their knowledge regardless of their experience level significantly increases the effectiveness of knowledge sharing practices, especially in environments involving software development (E10). However, there is a trade-off between removing strict hierarchies and maintaining the reporting lines required by the nature of large-scale organizations, and this needs to be addressed realistically while implementing such effective knowledge sharing initiatives (E9). These results suggest that applying a mixture of classical and flat structures as a hybrid approach can help to find the sweet spot to enable effective knowledge sharing in large-scale agile organizations.

6.3.6. Ad-Hoc Exchanges

Ad-hoc conversations are considered an important part of knowledge sharing and coordination practices by many studies and experts with varying degrees [56] [65] [67] [73] (E2, E3, E5, E6, E8, E9, E10, E11). As a mechanism, it enables high-quality communication and is important for fast decision-making [73]. In certain contexts, employees spend most of their time in ad-hoc meetings [65], coordinate most of their work with their teams via spontaneous meetings [73], and highly acknowledge the benefits of creating informal arenas for coordination [27]. Some studies further propose that ad-hoc meetings are the most crucial coordination mechanism in large-scale agile contexts because some scheduled meetings tend to have a status-reporting nature, such as Daily Standups, and the coordination part is usually neglected [56].

However, the results of our interview study challenge the generalizability of the findings of the systematic literature review regarding the significance of ad-hoc meetings over scheduled meetings. Our results confirm their importance as a coordination mechanism in large-scale agile environments but further suggest that they might not necessarily be more efficient than scheduled meetings (E3, E4, E5, E7, E8, E11). E3 compares ad-hoc exchanges and scheduled meetings as follows:

> *I believe the scheduled meetings are important. [...] If the team comes prepared for the scheduled meetings, the time can be reduced significantly. [...] Even Daily Scrum or Sprint Review would be a lot easier and smoother. [...] And in the ad-hoc part, it can be a distraction to the team if it happens constantly. [...] But it is also very beneficial if you have some issues that are happening to you and you cannot solve them without any help. You just need to go to that person without waiting for the next meeting. [...] And I believe (that) ad-hoc meetings have more purpose than scheduled meetings. So, go to solve that specific issue in 10 - 15 minutes and they solve it. But in the scheduled meetings, it is, more or less, more relaxed because you still have the next one.*

Along the same lines, several other experts also emphasize that both ad-hoc and scheduled meetings have their own benefits, and one cannot be picked over the other (E5, E6, E8, E11). Some experts even strictly prefer scheduled meetings over ad-hoc exchanges (E4, E7). E4 explains their reasoning about why scheduled meetings are more efficient as follows, with a concrete recent experience:

> *I think scheduled meetings are better, [...] because you can prepare better. [...] Since last week, I am doing a lot of knowledge transfer sessions for new developers, and I needed to explain today some very complex financial processes within the application. So, if someone would come to me and say, "Hey, now just in 30 minutes or 20 minutes, explain this stuff to me!" I would have explained, but it would probably take double the time because I would struggle a lot during the meeting. However, I spent yesterday one hour or so, I prepared for different scenarios (and) functionalities, [...] I searched for good test data, [...], I brought them to order, [...], and then the meeting was much more efficient. So I think I*

> *find [...], in terms of knowledge sharing, the scheduled and planned, prepared meetings much more efficient.*

Conversely, E10 sees the comparison from another view and argues that ad-hoc meetings are much preferable, especially from the software developers' perspective (E10). Scheduled meetings have the possibility of becoming redundant as there might not always be an agenda for the reserved time slot, but ad-hoc exchanges usually have a clear purpose defined beforehand (E10).

These diverse views, which are highly expected in cross-functional teams where different roles have varying interests as Ghobadi and Mathiassen [60] emphasize, regarding the comparison between ad-hoc and scheduled meetings highlight that the findings of our systematic literature review regarding the strict preference for ad-hoc meetings over scheduled ones are not generalizable in practice. Additionally, our results further show that the exact context of the ad-hoc meeting taking place is also a highly crucial factor in determining its efficiency. Therefore, although they are considered a valuable part of knowledge sharing and coordination practices [56] [65] [73] (E2, E3, E5, E6, E8, E9, E10, E11), their application might not always ensure efficiency. To reap the benefits of ad-hoc exchanges, another facilitator, namely the appropriate meeting selection that is discussed in Section 6.3.7 needs to be addressed so that the content of the intended exchange suits the nature of the ad-hoc meetings. In addition, the distraction factor should also be taken into account to achieve an effective realization (E1, E3, E11).

6.3.7. Appopriate Meeting Selection

Picking the right type of meeting and deciding whether that meeting is needed or not is considered essential for effective knowledge sharing and coordination by three of the interviewed experts (E1, E9, E10). The following quotes from E1, E9 respectively, summarize their view on this:

> *The efficiency of a meeting is always a question of expectation (of people) and what you expect. And you have to use the right type of meeting at the right moment.*

> *I think it is really important to have (scheduled meetings), but maybe not too many. So (we should) not have meetings (just) for the sake of having meetings, to be honest.*

E10 proposes an approach for deciding when to have a meeting or not, which is in line with Value 2 and Principle 7 of the Agile Manifesto [1] that prioritize working software:

> *If the cost of implementation is lower than the time of a meeting, we should not have a meeting. We should just have a prototype, we should just implement and see how it goes.*

As a result, putting a certain amount of effort into choosing the right type of meeting and eliminating the ones that are not required can significantly increase the effectiveness of knowledge sharing and coordination activities in large-scale agile organizations, as effective use of time is critical (E2, E6).

6.3.8. Encouraging Knowledge Sharing

One of the enabling factors discovered during the interview study is the encouragement of all team members to share their knowledge (E4, E11). E11 identifies three main reasons why people do not share their knowledge:

- Unawareness of one's own knowledge's significance
- Unwillingness to deal with the consequences
- Being uninterested in sharing knowledge in general because of seeing knowledge as a personal advantage

The second and third reasons are in line with the findings of Huber [78] who states that people tend to keep their knowledge to themselves when they think that sharing it will make them lose their relative advantage against their colleagues [78]. This highlights the importance of personal motives in knowledge sharing activities and the potential benefits of implementing certain incentives that encourage people to share their knowledge with their coworkers.

Yang & Wu [41] investigate the ways to promote knowledge sharing in organizations and propose a rewarding scheme to incentivize it. They conclude that rewarding individual knowledge sharing actions is more effective than periodic rewards adopted at the organizational level [41]. Along the lines of Yang & Wu [41], E11 suggests that making people proud of what they have is the key to tackling the three identified reasons for not sharing knowledge within the organizations (E11). To achieve this, a transparent work environment that encourages people to share their knowledge proudly is essential (E11).

E6 emphasizes the need for getting the courage to trust oneself to be able to interact with colleagues effectively (E6). E4 sees the encouragement of people to share their knowledge as a crucial factor for effective knowledge sharing and stresses that when people share knowledge, the other team members are also motivated to share their knowledge (E4). Therefore, initiating a company culture that triggers at least a certain group of people in the organization to freely share their knowledge can be scaled much faster with a snowball effect, and can facilitate knowledge sharing at a much higher pace than initially expected.

6.3.9. Clear Definition of Roles

When an organization attempts to scale a particular agile framework to a larger extent, the definition of the roles is not always clear [79]. However, defining clear roles is observed to be a facilitating factor for coordination in large-scale agile contexts (E7, E9). When the roles are unclear in certain meetings involving cross-functional members, participants often end up discussing the same topic repeatedly (E7). Similarly, members of large teams spend too much time answering the same questions that were already addressed previously [76]. Instead, synchronizing a common understanding of roles across the teams enables more effective coordination (E7).

Table 6.5 outlines the facilitators identified in the interview study and their effect on knowledge sharing and coordination practices.

Facilitator	Knowledge Sharing	Coordination
Documentation Efforts	✓	✓
Efficient Usage of Tools	✓	✓
Appropriate Participant Selection	✓	✓
Emotional Bonding	✓	✓
Flat Hierarchy	✓	
Ad-Hoc Exchanges	✓	✓
Appropriate Meeting Selection	✓	✓
Encouraging Knowledge Sharing	✓	
Clear Definition of Roles		✓

Table 6.5.: Identified facilitators in the interview study

7. Discussion

This chapter discusses the key findings of this thesis and describes its limitations.

7.1. Key Findings

This section summarizes the key findings of this thesis.

Knowledge sharing and coordination are tightly coupled, and a barrier or a facilitator affecting one of them is very likely to affect the other

In this thesis, we studied knowledge sharing and coordination simultaneously to find out through which mechanisms they take place and how they are affected by certain barriers and facilitators in large agile organizations. Investigating them parallelly enabled us to identify the patterns showing their correlation in certain contexts. Although the existing literature investigates the two individually and does not elaborate on their connection, the interview study of this thesis highlights that knowledge sharing and coordination are tightly coupled in large-scale agile contexts. Even though a mechanism such as a meeting, tool, or organizational structure might have a primary goal of supporting either knowledge sharing or coordination, the results show that most of the mechanisms also indirectly help unintended practices regarding knowledge sharing or coordination to occur. Likewise, a barrier or a facilitator affecting one of them is highly likely to affect the other. When asked about the barriers and facilitating factors of knowledge sharing and coordination, the experts mostly did not differentiate between the factors in terms of knowledge sharing and coordination and provided joint answers. This indicates that despite their unique definitions, both share a large set of common challenges and facilitators. This finding also remarks on the importance of keeping an eye on the other while investigating one of them, so that the unintended practices resulting from the other can also be taken into account.

Contrary to the findings of the existing literature, ad-hoc meetings are not necessarily more efficient than scheduled ones

The findings of the systematic literature review conducted as a part of this thesis show that ad-hoc meetings are proposed to be a more efficient mechanism than their scheduled counterparts [56] [67] [73]. Our interview study examined whether the results of the case studies regarding the efficiency of ad-hoc meetings are generalizable to broader contexts in practice. The results confirm the significance of ad-hoc exchanges for knowledge sharing and coordination but further demonstrate that they are not always more effective than the

scheduled meetings. In certain situations where a thorough preparation of the presenters is required to enable more effective knowledge sharing and coordination during a meeting, it is more beneficial to schedule the meeting with clear goals defined beforehand so that the presenters can use their time more effectively. In addition, ad-hoc meetings should be used in a way that ensures minimum disturbance to colleagues, as spontaneous requests are also reported to have a disturbing nature in certain contexts.

The implementation details of the online tools used for knowledge sharing and coordination are as important as the tools themselves, and the exact way they are used decides whether they work as effective mechanisms for knowledge sharing and coordination

The results of our study suggest that investigating the usage of online tools for knowledge sharing and coordination purposes is only meaningful when their exact implementations are taken into account. With inefficient setups, these online tools can substantially lose their beneficial functionalities, and reaping their benefits as knowledge sharing and coordination mechanisms can become significantly difficult. This also relates to the next key finding of this thesis as described below.

Although an online tool is not initially designed to be used for knowledge sharing or coordination purposes, it can still act as an important mechanism for knowledge sharing or coordination practices in large agile organizations

The previous key finding described above highlights the significance of the exact setup of online tools to be able to reap their benefits to their full potential. Our study further shows that even the tools that do not seem to be relevant for knowledge sharing and coordination practices at first sight, can become significantly useful mechanisms when they are used appropriately. These tools include design tools that enable knowledge sharing, as well as requirements management, version control, and repository tools such as Doors, Git, GitHub, and Bitbucket which facilitate both knowledge sharing and coordination. Therefore, the search for relevant practices while investigating knowledge sharing and coordination mechanisms should not be limited to well-known approaches in order to understand the full picture better.

Certain mechanisms whose primary goal is to facilitate knowledge sharing can indirectly enable effective coordination activities even if they are not intended to be doing so

The findings of this thesis reveal that certain knowledge sharing mechanisms facilitate coordination activities in large agile organizations. Agile Coffee Break, which is initially introduced as a knowledge sharing mechanism that helps colleagues to have a common understanding and awareness of what the others are doing, can indirectly mitigate the dependency awareness challenges [17] [68], and consequently enable more effective knowledge sharing practices. Similarly, COPs, whose main objective is to enable knowledge sharing across interested

individuals in a certain field, can serve as an indirect coordination mechanism by helping team members from various teams coordinate dependencies (E7).

Certain barriers and facilitators are inter-connected, and one of them can potentially create a snowball effect on the others, and consequently, can favor or hinder effective knowledge sharing and coordination even more than it would do just by itself

Our study identified several connections among the barriers and facilitators of knowledge sharing and coordination practices in large agile organizations. For instance, although the involvement of external colleagues is identified as a standalone barrier by itself, this thesis' results highlighted that it could further hinder transparency or even increase the existing lack of transparency in an organization (E5). Similarly, transparency, which is identified as a facilitating factor, enables a work environment where the members of the organization can feel proud of their knowledge and, therefore, feel encouraged to share their knowledge, which is another facilitator identified in this thesis (E11). Ad-hoc exchanges, which is one of the identified facilitators, require careful assessment before being chosen as a suitable mechanism for the given context (E4), which relates back to another identified facilitator, appropriate meeting selection. These connections can be critical for organizations and should be examined carefully, as the cascading nature might create a snowball effect, which can be highly beneficial if it is initiated by the facilitators, but can deteriorate the existing situation drastically if it originates from a barrier.

Besides the identified mechanisms and facilitators, subtle procedures and habits are also used to enable more effective knowledge sharing and coordination

Subtle procedures such as providing a link back to the feature in GitHub in the source code (E1), not closing the Sprints without finalizing the documentation even if all the tasks of the Sprint are completed (E4), and regularly adding needed documentation efforts into the personal task list (E6) are observed to be minor but effective mechanisms actively used by the experts in the industry to enable effective knowledge exchange and coordination.

Human factors and challenges resulting from remote work settings constitute a substantial part of the barriers to effective knowledge sharing and coordination

Regardless of which mechanisms and means are implemented, personal motives are observed to be crucial for knowledge sharing and coordination practices to take place effectively, as many experts report human-related factors as barriers emerging in various contexts. Similarly, remote work brings several interconnected challenges that are hard to completely eliminate with online tools.

7.2. Limitations

This section presents the limitations and threats to the validity of the results in this thesis.

The first limitation relates to the data collection process. Although this thesis is advised and supervised by two other researchers, without overlooking their contributions, it is important to highlight that this is an individually conducted study. This particularly affects the interview study, in which the number of interview partners and the length of the interviews depend highly on the number of researchers. By including more researchers in the study, it would be possible to conduct, transcribe, code, and analyze many more interviews, which would potentially help with the validity and the generalizability of the findings. Similarly, the number of existing publications that can be reviewed is also highly affected by the number of researchers in the study. With the inclusion of additional researchers, a wider range of publications could have been reviewed and analyzed to enlarge, support, and validate the current findings of the study.

The second limitation of this study arises from the limited time frame. Since this master's thesis had a strict start and due date, it differs from other studies in the sense that the available time for the data collection cannot be decided by the researcher; rather, it is pre-defined. For instance, as outlined in Chapter 3, several publications are based on years of on-site observations, or tens of expert interviews. Like the limited number of researchers, the limited time frame makes such an extensive data collection infeasible.

In addition, the interview study of this thesis has a potential limitation regarding the representativeness of the interview partners [80]. Possibly, conducting the interview series with a different set of experts might have produced different findings, which results in a potential validity threat.

However, various approaches were followed during the study to counter the validity threats mentioned above. To avoid the criticism that case studies face regarding the generalizability of their results [13], our interview study did not focus on a specific organization and enlarged the coverage by including many different organizations and roles. In addition, interviewees were encouraged to share their past experiences in large-scale agile environments besides their current organizations to counter the above-mentioned data collection limitations. Moreover, the conducted systematic literature review also helped counteract the validity threats that are inherent to the interview studies, as it provided the opportunity to validate certain findings by including the results from the existing research.

8. Conclusion and Future Work

The last chapter of this master's thesis summarizes the thesis and outlines potential future work.

8.1. Summary

In this master's thesis, mechanisms for effective knowledge sharing and coordination in large agile organizations are investigated. For this purpose, a systematic literature review and an interview study were performed in conjunction. The systematic literature review provided a rich set of case studies and highlighted the foundations for the interview study as well as certain key findings to be validated. Using these foundations, the interview study proceeded to identify the mechanisms for knowledge sharing and coordination, assessed the findings in the literature and their generalizability, and provided additional insights about the knowledge sharing and coordination mechanisms in large-scale agile environments. The interview study included 11 experts in total, covering 16 different roles as their past experiences were also investigated to mitigate the limitations of the study regarding the data collection phase. With this research approach, this thesis answered the following research questions.

> Research Question 1: How do knowledge exchange and coordination take place in large agile organizations, and which mechanisms are used for this purpose?

The interview study of this thesis identified 41 knowledge sharing and coordination mechanisms used in large agile organizations. These mechanisms were divided into three categories, namely meetings, organizational structures, and tools. The identified mechanisms included well-known, framework-specific mechanisms as well as company-specific approaches. Since one of the key findings of this thesis is the tightly coupled nature of knowledge sharing and coordination practices as explained in Chapter 7, the identified mechanisms were not classified further between knowledge sharing and coordination. Instead, each mechanism's primary goal was presented, as depicted in Table 6.1, Table 6.2, and Table 6.3.

> Research Question 2: What are the barriers to and facilitators for effective knowledge sharing and coordination in large agile organizations?

In the interview study of this thesis, 20 barriers, and 9 facilitators for effective knowledge sharing and coordination in large-scale agile organizations were identified. Our study revealed that challenging and enabling factors mostly tend to impact knowledge sharing and coordination practices in a similar way. Their impacts are classified as direct or indirect and depicted in Table 6.4 and Table 6.5.

Research Question 3: What are the benefits, trade-offs, and application contexts of knowledge exchange and coordination mechanisms in large agile organizations?

Our study highlighted the benefits, trade-offs, and application contexts of certain knowledge exchange and coordination practices in large-scale agile environments and presented them in conjunction with the identified barriers and facilitators. While scheduled meetings were revealed to be effective in contexts where thorough preparation before the meeting is preferable, ad-hoc meetings were identified as more suitable for solving problems on demand. For large-scale agile organizations, balancing this trade-off is crucial to ensure effective usage of ad-hoc meetings, and consequently, more efficient knowledge sharing and coordination practices. Similarly, formal meetings were identified to be crucial during the initial phases of team formation to establish some emotional bonding and team spirit, while more informal exchanges can replace them at the later stages. Furthermore, selecting the meetings appropriately is shown to be a crucial aspect to consider to facilitate effective knowledge sharing and coordination activities, which shows that the application contexts matter regardless of the mechanism in question.

8.2. Future Work

Despite our efforts to mitigate the limitations of this thesis, due to the limited time frame and the limited number of researchers, it was not possible to include more experts and organizations in the study. To have more extensive data to analyze, performing the interview study with a larger number of experts covering more organizations can be helpful to assess the generalizability of the results found in the literature even further, as well as to identify more mechanisms inherent to the large-scale agile organizations regarding knowledge sharing and coordination.

Although this thesis provides a variety of barriers to and facilitating factors, due to the limited time frame, the validation of the findings was not pursued further. Therefore, future research can focus on validating the barriers, facilitators, application contexts, and trade-offs identified in this study by assessing their applicability in different organizations and large-scale environments.

Building on the findings of this thesis, future work can focus on investigating the identified connections between the barriers and facilitators even further, and study their practical consequences in large agile organizations. In addition, the tightly coupled nature of the knowledge sharing and coordination practices can also be examined further to identify the environments in which they appear and their potential divergence in certain contexts.

A. Appendix

A.1. Interview Study Questions

A.1.1. General Questions about the Interview Partner

- What is your role at your current organization?
- How many years of experience do you have in your current role?
- Which other roles have you taken in large-scale agile organizations before?
- How many years of experience do you have in large-scale agile organizations in total?
- How many years of professional experience do you have in total?

A.1.2. Organization-Specific Questions

- What is the product that your teams working on?
- What is the size of your organization?
- What are the sizes of the teams in the large-scale agile context?
- Does your organization implement any large-scale agile framework? If yes, which one(s)?
- How is your organization structured?

A.1.3. Questions Regarding Knowledge Sharing and Coordination

- How do you share knowledge and coordinate in your large-scale agile setting?
- Which meetings do you use to share knowledge and coordinate in your large-scale agile setting?
- Which tools do you use to share knowledge and coordinate in your large-scale agile setting?
- What organizational structures do you use to share knowledge and coordinate in your large-scale agile setting?
- Which barriers do you observe in front of effective knowledge sharing and coordination in your large-scale agile setting?

- Which facilitators do you observe in front of effective knowledge sharing and coordination in your large-scale agile setting?
- How do you think the knowledge sharing and coordination practices can be improved in your large-scale agile setting?
- What is your opinion on working in a remote setting, and do you think it has any effect on knowledge sharing and coordination practices?
- What is your opinion on ad-hoc meetings, and do you think they are any different than scheduled meetings in terms of knowledge sharing and coordination efficiency?
- How do you handle dependencies in your large-scale agile setting?

List of Tables

Bibliography

[1] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, et al. *The agile manifesto*. 2001.

[2] M. Al-Zewairi, M. Biltawi, W. Etaiwi, A. Shaout, et al. "Agile software development methodologies: survey of surveys". In: *Journal of Computer and Communications* 5.05 (2017), p. 74.

[3] A. Mishra and Y. I. Alzoubi. "Structured software development versus agile software development: a comparative analysis". In: *International Journal of System Assurance Engineering and Management* 14.4 (2023), pp. 1504–1522.

[4] M. Van Oosterhout, E. Waarts, and J. Van Hillegersberg. "Change factors requiring agility and implications for IT". In: *European journal of information systems* 15 (2006), pp. 132–145.

[5] K. Preiss, S. L. Goldman, and R. N. Nagel. "Cooperate to compete: Building agile business relationships". In: *(No Title)* (1996).

[6] D. Reifer, F. Maurer, and H. Erdogmus. "Scaling agile methods". In: *IEEE Software* 20.4 (2003), pp. 12–14. DOI: 10.1109/MS.2003.1207448.

[7] K. Rolland, T. Dingsoyr, B. Fitzgerald, and K.-J. Stol. "Problematizing agile in the large: alternative assumptions for large-scale agile development". In: (2016).

[8] R. Hoda, P. Kruchten, J. Noble, and S. Marshall. "Agility in context". In: *Proceedings of the ACM international conference on Object oriented programming systems languages and applications*. 2010, pp. 74–88.

[9] P. Abrahamsson, K. Conboy, and X. Wang. *'Lots done, more to do': the current state of agile systems development research*. 2009.

[10] "What is SAFe?" In: (). URL: https://scaledagile.com/what-is-safe/.

[11] "LeSS Framework". In: (). URL: https://less.works/less/framework.

[12] K. Dikert, M. Paasivaara, and C. Lassenius. "Challenges and success factors for large-scale agile transformations: A systematic literature review". In: *Journal of Systems and Software* 119 (2016), pp. 87–108.

[13] R. K. Yin. *Case study research and applications*. Vol. 6. Sage Thousand Oaks, CA, 2018.

[14] M. Paasivaara, C. Lassenius, and V. T. Heikkilä. "Inter-team coordination in large-scale globally distributed scrum: Do scrum-of-scrums really work?" In: 2012, pp. 235–238. DOI: 10.1145/2372251.2372294. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84867504349&doi=10.1145%2f2372251.2372294&partnerID=40&md5=c56a9fe132a4c15a6b9c360a21333657.

[15] S. Keele et al. *Guidelines for performing systematic literature reviews in software engineering*. 2007.

[16] M. D. Myers and M. Newman. "The qualitative interview in IS research: Examining the craft". In: *Information and organization* 17.1 (2007), pp. 2–26.

[17] S. Bick, K. Spohrer, R. Hoda, A. Scheerer, and A. Heinzl. "Coordination Challenges in Large-Scale Software Development: A Case Study of Planning Misalignment in Hybrid Settings". In: *IEEE Transactions on Software Engineering* 44.10 (2018), pp. 932–950. DOI: 10.1109/TSE.2017.2730870. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85028940218&doi=10.1109%2fTSE.2017.2730870&partnerID=40&md5=b37898dba426c6678e5c5a180821b461.

[18] J. A. Highsmith. *Agile software development ecosystems*. Addison-Wesley Professional, 2002.

[19] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta. *Agile Software Development Methods: Review and Analysis*. 2017. arXiv: 1709.08439 [cs.SE].

[20] M. Hron and N. Obwegeser. "Scrum in practice: an overview of Scrum adaptations". In: *Hawaii International Conference on System Sciences*. Curran Associates, Inc. 2018, pp. 4496–4505.

[21] T. Dingsøyr, S. Nerur, V. Balijepally, and N. Moe. "A decade of agile methodologies: Towards explaining agile software development". In: *Journal of Systems and Software* 85 (June 2012), pp. 1213–1221. DOI: 10.1016/j.jss.2012.02.033.

[22] K. Schwaber. "SCRUM Development Process". In: *Business Object Design and Implementation*. Ed. by J. Sutherland, C. Casanave, J. Miller, P. Patel, and G. Hollowell. London: Springer London, 1997, pp. 117–134. ISBN: 978-1-4471-0947-1.

[23] K. Schwaber and M. Beedle. *Agile Software Development with Scrum*. 1st. USA: Prentice Hall PTR, 2001. ISBN: 0130676349.

[24] K. Schwaber. *Agile project management with Scrum*. Microsoft press, 2004.

[25] K. Schwaber and J. Sutherland. "The Scrum Guide 2020". In: (2020). URL: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf.

[26] T. Dingsøyr and N. B. Moe. "Towards Principles of Large-Scale Agile Development: A Summary of the workshop at XP2014 and a revised research agenda". In: *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation: XP 2014 International Workshops, Rome, Italy, May 26-30, 2014, Revised Selected Papers 15*. Springer. 2014, pp. 1–8.

[27] T. Dingsøyr, N. B. Moe, T. E. Fægri, and E. A. Seim. "Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation". In: *Empirical Software Engineering* 23.1 (2018), pp. 490–520. DOI: 10.1007/s10664-017-9524-2. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85020136655&doi=10.1007%2fs10664-017-9524-2&partnerID=40&md5=85febfd2f043293a7a9b73bacb9b3cd7.

[28] "Scaling Agile at Spotify". In: (2012).

[29] J. Sutherland and S. Inc. "The Scrum at Scale Guide". In: (2022).

[30] "Welcome to Scaled Agile Framework® 6.0!" In: (). URL: https://scaledagileframework.com/about/.

[31] "SAFe Lean-Agile Principles". In: (). URL: https://scaledagileframework.com/safe-lean-agile-principles/.

[32] "SAFe 6.0". In: (). URL: https://scaledagileframework.com/.

[33] "Agile Release Train". In: (). URL: https://scaledagileframework.com/agile-release-train/.

[34] "Agile Release Train". In: (). URL: https://scaledagileframework.com/solution-train/.

[35] K. Schwaber. *Agile project management with Scrum*. Microsoft press, 2004.

[36] Ö. Uludag, M. Kleehaus, C. Caprano, and F. Matthes. "Identifying and structuring challenges in large-scale agile development based on a structured literature review". In: *2018 IEEE 22nd international enterprise distributed object computing conference (EDOC)*. IEEE. 2018, pp. 191–197.

[37] M. Ipe. "Knowledge Sharing in Organizations: A Conceptual Framework". In: *Human Resource Development Review* 2.4 (2003), pp. 337–359. DOI: 10.1177/1534484303257985. eprint: https://doi.org/10.1177/1534484303257985. URL: https://doi.org/10.1177/1534484303257985.

[38] R. Farooq. "A conceptual model of knowledge sharing". In: *International Journal of Innovation Science* 10.2 (2018), pp. 238–260.

[39] T. Zheng. "A literature review on knowledge sharing". In: *Open Journal of Social Sciences* 5.3 (2017), pp. 51–58.

[40] E. Helmstädter. "The economics of knowledge sharing: A new institutional approach". In: *(No Title)* (2003).

[41] H.-L. Yang and T. C. Wu. "Knowledge sharing in an organization". In: *Technological Forecasting and Social Change* 75.8 (2008), pp. 1128–1156. ISSN: 0040-1625. DOI: https://doi.org/10.1016/j.techfore.2007.11.008. URL: https://www.sciencedirect.com/science/article/pii/S0040162507002028.

[42] W. Tsai. "Social structure of "coopetition" within a multiunit organization: Coordination, competition, and intraorganizational knowledge sharing". In: *Organization science* 13.2 (2002), pp. 179–190.

[43] C. Moorman and A. S. Miner. "Organizational improvisation and organizational memory". In: *Academy of management Review* 23.4 (1998), pp. 698–723.

[44] W. R. King. "Knowledge sharing". In: *Encyclopedia of Knowledge Management, Second Edition*. IGI Global, 2011, pp. 914–923.

[45] K. Beck. *Extreme programming explained: embrace change*. addison-wesley professional, 2000.

[46] K. Schwaber and M. Beedle. *Agile software development with Scrum*. Prentice Hall PTR, 2001.

[47] T. Chau and F. Maurer. "Knowledge sharing in agile software teams". In: *Logic versus approximation: essays dedicated to Michael M. Richter on the occasion of his 65th birthday*. Springer, 2004, pp. 173–183.

[48] V. Santos, A. Goldman, and C. R. B. de Souza. "Fostering effective inter-team knowledge sharing in agile software development". In: *Empirical Software Engineering* 20.4 (2015), pp. 1006–1051. DOI: 10.1007/s10664-014-9307-y. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84930482361&doi=10.1007%2fs10664-014-9307-y&partnerID=40&md5=ea12c34e9175d64b6a262a61a7f73d6e.

[49] K. Kuusinen, P. Gregory, H. Sharp, L. Barroca, K. Taylor, and L. Wood. "Knowledge sharing in a large agile organisation: A survey study". In: *Lecture Notes in Business Information Processing* 283 (2017), pp. 135–150. DOI: 10.1007/978-3-319-57633-6_9. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85018699061&doi=10.1007%2f978-3-319-57633-6_9&partnerID=40&md5=40016d9cf503f0c4410b24c0015dd83a.

[50] N. Bernstein. "The coordination and regulation of movements". In: *(No Title)* (1967).

[51] J. D. Mooney. *Principles of Organization*. Harper and Brothers Publishers, 1947.

[52] A. H. Van de Ven, A. L. Delbecq, and R. Koenig Jr. "Determinants of coordination modes within organizations". In: *American sociological review* (1976), pp. 322–338.

[53] C. Camerer and M. Knez. "Coordination in organizations: A game-theoretic perspective". In: *Organizational decision making* (1997), pp. 158–188.

[54] B. Curtis, H. Krasner, and N. Iscoe. "A field study of the software design process for large systems". In: *Communications of the ACM* 31.11 (1988), pp. 1268–1287.

[55] R. E. Kraut and L. A. Streeter. "Coordination in software development". In: *Communications of the ACM* 38.3 (1995), pp. 69–82.

[56] H. Nyrud and V. Stray. "Inter-team coordination mechanisms in large-scale agile". In: vol. Part F129907. 2017. DOI: 10.1145/3120459.3120476. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85029844460&doi=10.1145%2f3120459.3120476&partnerID=40&md5=6b29980b9c2e1850279c82d64e0e53f3.

[57] V. Stray, N. B. Moe, and M. Noroozi. "Slack Me if You Can! Using Enterprise Social Networking Tools in Virtual Agile Teams". In: 2019, pp. 111–121. DOI: 10.1109/ICGSE.2019.00031. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85072111140&doi=10.1109%2fICGSE.2019.00031&partnerID=40&md5=847f78c156e4387fc55a50208589a590.

[58] S. Dorairaj, J. Noble, and P. Malik. "Knowledge management in distributed agile software development". In: 2012, pp. 64–73. DOI: 10.1109/Agile.2012.17. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84868286498&doi=10.1109%2fAgile.2012.17&partnerID=40&md5=9271e6cdd257c64e1928f61609141dc0.

[59] M. Berntzen, R. Hoda, N. B. Moe, and V. Stray. "A Taxonomy of Inter-Team Coordination Mechanisms in Large-Scale Agile". In: *IEEE Transactions on Software Engineering* 49.2 (2023), pp. 699–718. DOI: 10.1109/TSE.2022.3160873. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85127075670&doi=10.1109%2fTSE.2022.3160873&partnerID=40&md5=742e23dfab8df66814682eb45c2a03a1.

[60] S. Ghobadi and L. Mathiassen. "Perceived barriers to effective knowledge sharing in agile software teams". In: *Information systems journal* 26.2 (2016), pp. 95–125.

[61] B. Kitchenham. "Procedures for undertaking systematic reviews: Joint technical report". In: *Computer Science Department, Keele University (TR/SE-0401) and National ICT Australia Ltd.(0400011T. 1)* (2004).

[62] C. G. Cobb. *The project manager's guide to mastering Agile: Principles and practices for an adaptive approach*. John Wiley & Sons, 2023.

[63] J. Saldaña. "The coding manual for qualitative researchers". In: (2021).

[64] A. Huberman et al. "Qualitative data analysis a methods sourcebook". In: (2014).

[65] V. Stray. "Planned and unplanned meetings in large-scale projects". In: vol. Part F147763. 2018. DOI: 10.1145/3234152.3234178. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85065770966&doi=10.1145%2f3234152.3234178&partnerID=40&md5=bfc7812b186618f9bc05202f8edbf188.

[66] V. Stray, A. Bai, N. Sverdrup, and H. Mork. "Empowering agile project members with accessibility testing tools: a case study". In: *Agile Processes in Software Engineering and Extreme Programming: 20th International Conference, XP 2019, Montréal, QC, Canada, May 21–25, 2019, Proceedings 20*. Springer International Publishing. 2019, pp. 86–101.

[67] N. B. Moe, T. Dingsøyr, and K. Rolland. "To schedule or not to schedule? An investigation of meetings as an inter-team coordination mechanism in large-scale agile software development". In: *International Journal of Information Systems and Project Management* 6.3 (2018), pp. 45–59. DOI: 10.12821/ijispm060303. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85055290295&doi=10.12821%2fijispm060303&partnerID=40&md5=357ec528eb1c0057c226dd22c2186967.

[68] T. Gustavsson. "Changes Over Time in a Planned Inter-team Coordination Routine". In: *Lecture Notes in Business Information Processing* 364 (2019), pp. 105–111. DOI: 10.1007/978-3-030-30126-2_13. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85072840633&doi=10.1007%2f978-3-030-30126-2_13&partnerID=40&md5=c1c0cdc1b6afe8b03e9a2e931231c864.

[69] M. Paasivaara and C. Lassenius. "Empower Your Agile Organization: Community-Based Decision Making in Large-Scale Agile Development at Ericsson". In: *IEEE Software* 36 (Mar. 2019), pp. 64–69. DOI: 10.1109/MS.2018.2886827.

[70] F. Almeida, E. Miranda, and J. Falcão. "Challenges and facilitators practices for knowledge management in large-scale scrum teams". In: *Journal of Information Technology Case and Application Research* 21.2 (2019), pp. 90–102. DOI: 10.1080/15228053.2019.1637087. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85068619907&doi=10.1080%2f15228053.2019.1637087&partnerID=40&md5=ccf13d31370d0c6437d3356a97feb8b3.

[71] M. Dahlqvist and J. Forsberg. "Inter-team knowledge sharing : A case study on co-located teams' drivers and barriers for KS". In: 2018. URL: https://api.semanticscholar.org/CorpusID:49525188.

[72] J. Gifford. "Remote working: unprecedented increase and a developing research agenda". In: *Human Resource Development International* 25.2 (2022), pp. 105–113. DOI: 10.1080/13678868.2022.2049108. eprint: https://doi.org/10.1080/13678868.2022.2049108. URL: https://doi.org/10.1080/13678868.2022.2049108.

[73] M. Berntzen, N. B. Moe, and V. Stray. "The product owner in large-scale agile: An empirical study through the lens of relational coordination theory". In: *Lecture Notes in Business Information Processing* 355 (2019), pp. 121–136. DOI: 10.1007/978-3-030-19034-7_8. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85065885599&doi=10.1007%2f978-3-030-19034-7_8&partnerID=40&md5=80dbbaa4fd340c3b63e32a09763407be.

[74] T. Sporsem and N. B. Moe. "Coordination Strategies When Working from Anywhere: A Case Study of Two Agile Teams". In: *Lecture Notes in Business Information Processing* 445 LNBIP (2022), pp. 52–61. DOI: 10.1007/978-3-031-08169-9_4. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85132983332&doi=10.1007%2f978-3-031-08169-9_4&partnerID=40&md5=81da3ac6b2ba1599fe0dd67f630483e6.

[75] F. O. Bjørnson, J. Wijnmaalen, C. J. Stettina, and T. Dingsøyr. "Inter-team coordination in large-scale agile development: A case study of three enabling mechanisms". In: *Lecture Notes in Business Information Processing* 314 (2018), pp. 216–231. DOI: 10.1007/978-3-319-91602-6_15. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85048042273&doi=10.1007%2f978-3-319-91602-6_15&partnerID=40&md5=42b3c42ff7679cffeb27cc7a7b1fdb31.

[76] H. Holz and F. Maurer. "Knowledge management support for distributed agile software processes". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 2640 (2003), pp. 60–80. DOI: 10.1007/978-3-540-40052-3_7. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0348216558&doi=10.1007%2f978-3-540-40052-3_7&partnerID=40&md5=de9568c88b93cf92c275af40d0e51f43.

[77] L. Rising and N. Janoff. "The Scrum software development process for small teams". In: *IEEE Software* 17.4 (2000), pp. 26–32. DOI: 10.1109/52.854065.

[78] G. P. Huber. "Transfer of knowledge in knowledge management systems: unexplored issues and suggested studies". In: *European Journal of Information Systems* 10.2 (2001), pp. 72–79. DOI: 10.1057/palgrave.ejis.3000399. eprint: https://doi.org/10.1057/palgrave.ejis.3000399. URL: https://doi.org/10.1057/palgrave.ejis.3000399.

[79] H. Edison, X. Wang, and K. Conboy. "Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review". In: *IEEE Transactions on Software Engineering* PP (Mar. 2021), pp. 1–1. DOI: 10.1109/TSE.2021.3069039.

[80] C. B. Seaman. "Qualitative methods in empirical studies of software engineering". In: *IEEE Transactions on software engineering* 25.4 (1999), pp. 557–572.