The Journal of Systems and Software 230 (2025) 112561

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

It all starts with structure: investigating learning dynamics in large-scale agile software development

Muhammad Ovais Ahmad [a,#], Hadi Ghanbari [b,#,*], Tomas Gustavsson [c,#], Bikesh Raj Upreti [d,#]

[a] Department of Computer Science, Karlstad University, Sweden
[b] Department of Information and Service Management, Aalto University School of Business, Ekonominaukio 1, 02150, Espoo, Finland
[c] Karlstad Business School, Karlstad University, Sweden
[d] UQ Business School, The University of Queensland, Australia

ARTICLE INFO

Edited by: Dr. Christoph Treude

Keywords:
Agile software development
Large-scale agile
Team structure
Team learning
Team cohesion
Knowledge sharing
Reflexivity
Structural equation modeling

ABSTRACT

Agile software development (ASD) methods have increasingly been used in large-scale software development projects. While ASD emphasizes the importance of social interactions between practitioners for continuous reflection and knowledge sharing, these learning activities become incredibly challenging in large-scale projects. Drawing on well-established theoretical concepts, we posit that learning in large-scale ASD projects requires a suitable environment that empowers practitioners to openly and frequently engage in social interactions, which are essential for reflection and knowledge sharing. We hypothesize that several team-level factors shape individuals' perceptions about the learning environment and learning activities in their projects, ultimately influencing their learning behavior. To test our model, we collected survey responses from practitioners working in large-scale ASD projects in five Swedish companies ($N = 159$). The data was analyzed using confirmatory factor analysis (CFA) and structural equation modeling (SEM). The results show that team structure plays a crucial role in promoting team cohesion and reflexivity, which, alongside knowledge sharing, contribute to the learning process. Our study provides ASD development research with a theoretically informed understanding of the interrelationship between the learning environment and learning activities in large-scale ASD. Our results guide practitioners in fostering suitable learning environments and enhancing learning in large-scale ASD.

1. Introduction

Agile software development (ASD) methods emphasize the importance of continuous value delivery and responsiveness to change (Conboy, 2009), frequent social interactions and communications (Hennel and Rosenkranz, 2021; Uludağ et al., 2022), and continuous feedback and reflection (Peeters et al., 2022) within small self-organizing and independent teams. Considering the various benefits of ASD in small projects (Dikert et al., 2016), more recently, organizations have moved towards using ASD methods in large-scale projects (Carroll et al., 2023). Consequently, a number of methods and frameworks, such as the Scaled Agile Framework (SAFe) (Knaster and Leffingwell, 2018) and Large-Scale Scrum (LeSS) (Larman and Vodde, 2010), have been proposed and used by organizations in large-scale projects (Uludağ et al., 2022). However, using these methods in large-scale projects is shown to be more difficult, to the extent that some have called large-scale ASD "a thorny set of issues" (Reifer et al., 2003). While there is a lack of "firm agreement as to what constitutes large-scale development" (Edison et al., 2022, p. 2709), according to a widely used definition proposed by (Dikert et al., 2016), a large-scale project refers to a project "with 50 or more people or at least six teams" (p. 88).

Due to the size and complexity of large-scale ASD projects, interpersonal interactions and collaborations become challenging and it becomes difficult to align and coordinate multiple development teams working on different parts of a project or product (Carroll et al., 2023; Edison et al., 2022). Therefore, large-scale ASD faces a variety of social issues related to team and project management, inter-team

* Corresponding author.
E-mail addresses: Ovais.ahmad@kau.se (M.O. Ahmad), hadi.ghanbari@aalto.fi (H. Ghanbari), tomas.gustavsson@kau.se (T. Gustavsson), b.upreti@uq.edu.au (B.R. Upreti).
[#] The authors are listed in alphabetic order.

https://doi.org/10.1016/j.jss.2025.112561
Received 17 June 2024; Received in revised form 15 April 2025; Accepted 3 July 2025
Available online 6 July 2025

collaboration, and customer collaboration (Edison et al., 2022), as well as learning and its underlying activities (Berntzen et al., 2023; Dingsøyr et al., 2023). The latter is especially problematic because establishing an environment, in which software professionals engage in learning, is known to be a prerequisite for enhancing software development practices (van Solingen et al., 2000) and an organization's overall learning and continuous improvement (Qumer and Henderson-Sellers, 2008) which is central to ASD (Berntzen et al., 2023).

Agile methods emphasize the importance of continuous learning in understanding how to adapt and become more effective in achieving project goals (Babb et al., 2014). Learning is an interpersonal process that unfolds as software professionals engage in knowledge sharing and reflection (Buvik and Tkalich, 2022; Ngereja and Hussein, 2021; Peeters et al., 2022). Through knowledge sharing, they "share their work-related experience, expertise, know-how, and contextual information with other colleagues" (Lin, 2007), while reflection enables them to review and learn from their past experiences (Peeters et al., 2022). Knowledge sharing over longer periods of time, as well as reflection, are shown to be crucial for facilitating learning and achieving continuous improvement in large-scale ASD (Kasauli, 2021). ASD relies heavily on interpersonal relationships and social interactions that enable knowledge sharing and reflection among software professionals (Hennel and Rosenkranz, 2021; Peeters et al., 2022; Strode et al., 2022). While traditional ASD methods are designed to facilitate such social interactions in small projects (Goh et al., 2013; Uludağ et al., 2022), they do not offer sufficient guidance for large-scale projects (Uludağ et al., 2022). Thus, when ASD methods are used at scale, learning becomes challenging (Carroll et al., 2023; Edison et al., 2022). Previous studies show that in working environments that foster social interactions between developers (Gupta et al., 2019) and continuous learning (Babb et al., 2014), for instance, through agile practices like daily standups and retrospective meetings, individuals are encouraged to actively engage in communication and information sharing within and between teams. Drawing on these studies, we argue that effective reflection and knowledge sharing in large-scale ASD necessitate a supportive environment in which software professionals are encouraged to engage in social interactions (Hennel and Rosenkranz, 2021; Thorgren and Caiman, 2019) and empowered to ask questions and share ideas (Babb et al., 2014; Garvin et al., 2008). As a result, experiences and knowledge gained by individuals are shared and integrated within teams and across teams (Mehta et al., 2022), enabling them (Babb et al., 2014; Buvik and Tkalich, 2022; Ozer and Vogel, 2015; Peeters et al., 2022).

A recent systematic mapping study (Uludağ et al., 2022) shows that previous literature on large-scale ASD mostly focused on agile practices at scale, communication and coordination in large-scale projects, and scaling agile frameworks. However, despite its importance, learning in the ASD context remains an understudied area (Mehta et al., 2022). We still lack empirical studies that provide us with a theoretically grounded understanding of learning in large-scale ASD, which has been reported to be problematic (Dingsøyr et al., 2023; Edison et al., 2022). We especially lack a scientific understanding of the factors that shape software professionals' perceptions about the learning environment and engagement in learning activities in large-scale ASD projects, which is known to impact an organization's overall learning (Qumer and Henderson-Sellers, 2008). This is problematic considering the importance of team learning and its impact on team performance, project success, and innovation capabilities in ASD projects (Dingsøyr et al., 2023, 2016; Mehta et al., 2022; Ngereja and Hussein, 2021; Ozer and Vogel, 2015; Strode et al., 2022). As a first step in addressing this gap, we set out to examine the following question:

RQ1: What are the factors that shape the learning environment and influence learning behavior in large-scale ASD?

To answer this question, drawing on well-established concepts proposed in the extant literature, we hypothesized that in large-scale ASD projects, a learning environment is shaped by team structure and the way tasks and responsibilities are perceived by individuals. Appropriate team structures, with clearly defined roles, objectives, and clear priorities (Bresman and Zellmer-Bruhn, 2013) improve interactions between team members (Dey and MP, 2020; Kakar, 2016) and enhance team cohesion (Brawley et al., 1993; Callow et al., 2009; Widmeyer and Ducharme, 1997). In turn, such well-structured and cohesive environments facilitate learning as they enable team members to better engage in knowledge sharing (Bresman and Zellmer-Bruhn, 2013; Bunderson and Boumgarden, 2010) and reflection (Hennel and Rosenkranz, 2021; Schippers et al., 2008). To test our theoretical model, we used an online survey to collect data from software professionals working in large-scale ASD projects in Sweden. We evaluated the measurement model following confirmatory factor analysis (CFA) and proceeded with structural equation modeling (SEM) to test our hypotheses. Our results supported all the hypothesized interrelationships except for that between knowledge sharing and team cohesion.

2. Research background

In their systematic literature review, Edison et al. (2022) report a variety of issues related to the implementation of ASD methods in large-scale projects, including issues related to team and project management, inter-team collaboration, and customer collaboration. These issues are known to arise from various factors, including a large number of teams, roles, and personalities; unknown team composition; diverse and competing agendas between teams; and complex and often unclear goals (Conboy and Carroll, 2019; Edison et al., 2022; Reifer et al., 2003). Another recent systematic mapping study by Uludağ et al. (2022) examines the state of the art of large-scale ASD research and identified ten research streams (Uludağ et al., 2022). These literature reviews show that previous empirical studies on large-scale ASD have mostly focused on reporting lessons learned from agile transformation processes and studying the coordination and collaboration in large-scale ASD projects (Paasivaara et al., 2018; Uludağ et al., 2022).

A group of studies has examined the challenges and success factors of following the most popular large-scale ASD methods and frameworks, such as SAFe, LeSS, and Scrum-at-Scale (see Edison et al., 2022). For instance, Putta et al. (2018) and Kalenda et al. (2018) examined the challenges and success factors of Scaled Agile Framework (SAFe) and Large Scale Scrum (LeSS). Other studies, mostly qualitative case studies, have examined the collaboration and coordination challenges associated with large-scale ASD projects. For instance, in a longitudinal case study, Berntzen et al. (2023) examined coordination challenges in large-scale ASD. Through their field study, Dingsøyr, Moe, et al. (2018) find customer involvement, software architecture, and inter-team coordination as the key challenging areas in large-scale ASD. Another study by Dingsøyr et al. (2023) reports inter-team coordination as a key challenge in large-scale projects and proposes a set of strategies for enhancing coordination mechanisms.

Establishing an environment in which software professionals engage in learning is a prerequisite for enhancing software development practices (van Solingen et al., 2000) and organizations' overall learning and performance (Qumer and Henderson-Sellers, 2008). Learning consists of activities through which teams codify and convert tacit knowledge into explicit knowledge (Gibson and Vermeulen, 2003). Learning in software development teams is influenced by both a team's internal knowledge resources (i.e., the skills and knowledge of each member) and external knowledge resources in the form of the skills and capabilities available outside the team (Mehta et al., 2022). Therefore, continuous learning is crucial for software teams because it enables them to be more reflexive and adaptive while improving the level of knowledge and skills produced collectively by members (Dingsøyr et al., 2016; Ellis et al., 2003; Müller et al., 2009). However, previous studies show that learning and its underlying activities, such as knowledge sharing, which heavily rely on communication and social interaction between software professionals, become problematic in large-scale ASD (Dingsøyr et al., 2023; Edison et al., 2022). In particular, knowledge sharing is shown to be

challenging in large-scale projects due to the larger number of teams and team members (Goh et al., 2013; Kalenda et al., 2018) and more complex knowledge dependencies (Lindsjørn et al., 2018; Strode et al., 2022).

Learning is fundamental to software development projects, and there is a high demand for both individual and team learning in large-scale ASD (Britto et al., 2016). Learning and its underlying activities help teams create a shared mental model (Mathieu et al., 2008). Considering the social and interpersonal complexities of large-scale ASD, having a shared understanding of the project and its objectives is crucial for teams and team members, not only to increase their skills but also to improve their performance (Dingsøyr et al., 2016). In particular, creating a shared mental model is shown to enhance team cohesion and facilitate team learning (Liu et al., 2023). It is important that team members are encouraged and empowered to engage in reflection and knowledge sharing activities that heavily rely on social interactions. To this end, ASD emphasizes the importance of interpersonal relationships and social interactions (Strode et al., 2022) that enable knowledge sharing and cooperative learning (Mehta et al., 2022), which occurs through consensus building and cooperation between team members (Janz and Prasarnphanich, 2009). By creating a suitable environment for knowledge sharing and continuous learning in large-scale ASD projects, companies can ensure that developers are able to perform their tasks (Mustaqim et al., 2023). For instance, facilitating frequent communication in large-scale ASD is reported to lower the barriers to asking questions and sharing knowledge, which in turn enhances learning (Liu et al., 2023). In addition, Another recent study shows that the formation of high-quality relationships enhances a team's capacity to learn from failures in large-scale ASD, which in turn enhances their outcomes (Ahmad, 2025b). As such, the author argues that promoting a culture of continuous learning is instrumental in improving team performance in large-scale ASD.

Sharing and managing knowledge over longer periods is crucial for facilitating learning in large-scale ASD (Britto et al., 2016; Kasauli et al., 2021), as the knowledge gained by individuals is shared and integrated across teams (Mehta et al., 2022). This allows software professionals and teams to reflect on their tasks and responsibilities (Babb et al., 2014), explore product feasibility, and enable continuous improvement through reflection (Kasauli et al., 2021). For instance, while retrospectives are beneficial for addressing small intra-team related issues, conducting project-level retrospectives enhances learning throughout large-scale ASD projects (Dingsøyr, Mikalsen, et al., 2018). The authors suggest that large-scale ASD projects should devote more time to retrospective meetings (Dingsøyr, Mikalsen, et al., 2018). In their case study, Paasivaara and Lassenius (2014) show that using communities of practice in distributed large-scale ASD projects facilitates coordination and enhances knowledge sharing and learning. Another qualitative study (Ahmad, 2025a) reports that by organizing structured knowledge-sharing sessions and encouraging experimentation and reflection, companies can promote team learning in large-scale ASD projects. Finally, Liu et al. (2023) report that utilizing coordination mechanisms such as multiplayer games and daily quizzes enhances team cohesion and team learning.

One might argue that learning in large-scale ASD is similar to small-scale ASD projects. However, the dynamics and interdependencies within large-scale ASD projects are different from other work contexts and small-scale ASD projects for at least two reasons. First, ASD was originally proposed as a suitable approach for small projects which rely on intra-team face-to-face interactions and self-management as well as simplicity of development processes (Peeters et al., 2022). Large-scale ASD projects, however, consist of a large number of social and technological interdependencies (Carroll et al., 2023; Uludağ et al., 2022) which makes interaction and collaboration challenging (Edison et al., 2022) and leads to complex knowledge boundaries (Lindsjørn et al., 2018). Therefore, when ASD methods are implemented at scale, there is a need for extra connecting practice (Edison et al., 2022) and formal coordination and communication (Uludağ et al., 2022). While these practices can help with organizing and coordinating inter-team collaboration, they can cause overhead (Dikert et al., 2016), which might reduce the opportunities for informal social interactions and learning.

Second, ASD promotes iterative and short product turnarounds (Peeters et al., 2022) by emphasizing shorter development and delivery horizons where the product content and their priorities can change even at the very late stages (Dikert et al., 2016). This becomes problematic in large-scale ASD projects, where software professionals must often interact and closely collaborate with other non-software development teams and units. The differences between these teams' perspectives about the iterative, flexible, and short development and release cycles can cause friction between teams and dissatisfaction within teams and hinder information sharing between teams (Dikert et al., 2016). As a result, these issues can reduce opportunities for knowledge sharing, reflection, and learning in large-scale ASD projects.

3. Hypothesis formulation and theoretical model

We develop a theoretical framework based on well-established theoretical concepts originally proposed in the social psychology and organizational science literature and used in software engineering studies (see Fig. 1). These concepts and their hypothesized relationships are discussed in the following sections.

3.1. Team structure

Team structure falls under the umbrella of team design and focuses on task specialization, hierarchical arrangements, and formalizing team objectives and procedures (Bresman and Zellmer-Bruhn, 2013). Team structure depends on the needs of a team and the particular context or specialization (Bresman and Zellmer-Bruhn, 2013). The characteristics of an appropriate team structure include clearly defined roles and task controls, clear objectives and goals, and clear priorities and norms (Bresman and Zellmer-Bruhn, 2013). Teams with such a structure provide opportunities for more interaction between team members (Dey and MP, 2020) and, consequently, more opportunities for sharing information and team learning (Bresman and Zellmer-Bruhn, 2013; Bunderson and Boumgarden, 2010; Gladstein, 1984).

A better team structure is suggested to reduce the frequency of conflicts within teams (i.e., cohesiveness) while increasing knowledge sharing and exchange among team members (Bresman and Zellmer-Bruhn, 2013; Bunderson and Boumgarden, 2010). Setting clear goals and objectives provides teams with a clear focus and enhances intergroup communication (Widmeyer and Ducharme, 1997), while a lack of shared understanding about goals and priorities leads to inter-team coordination problems in large-scale projects (Berntzen et al., 2022, 2023). In addition, promoting the acceptance of those goals and teamwork within teams enhances team cohesion (Brawley et al., 1993; Callow et al., 2009; Widmeyer and Ducharme, 1997). The existing literature also suggests that team structure plays an important role in team learning (Bresman and Zellmer-Bruhn, 2013; Bunderson and Boumgarden, 2010).

Previous research suggests that team structure contributes to agility (Langholf and Wilkens, 2021), reflexivity, and learning (Hennel and Rosenkranz, 2021; Mehta et al., 2022) in ASD teams. A case study conducted in a German firm (Langholf and Wilkens, 2021) shows that team structure enhances agility in software teams. Another multiple case study (Hennel and Rosenkranz, 2021) proposes that team structure helps improve learning and reflexivity in ASD teams (Hennel and Rosenkranz, 2021). However, when ASD methods are implemented at scale, there is a need for extra connecting practice (Edison et al., 2022) and formal procedures (Uludağ et al., 2022), to help with organizing and coordinating inter-team collaborations and defining team objectives and responsibilities (Mehta et al., 2022).

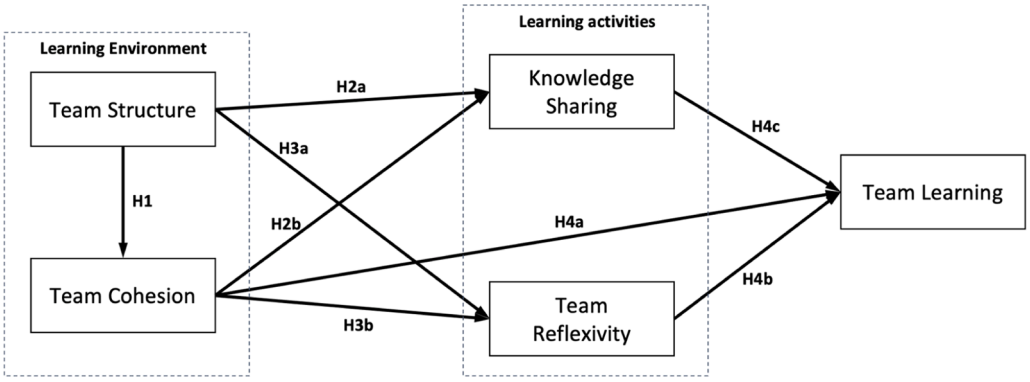

Fig. 1. The proposed theoretical model.

3.2. Team cohesion

Team cohesion refers to the extent to which team members have a favorable view of one another, identify themselves positively with the team, and express a willingness to remain as a member of the team (Shaw, 1981). Typically, team cohesion is described as a combination of task and social cohesiveness; task cohesion relates to a team's collective skills and commitment to performing its tasks and achieving its goals, whereas social cohesion relates to the social bonds and relationships within the team (Dey and MP, 2020). Through social integration, team cohesion promotes a sense of community among members of ASD teams to align them with the team's shared objectives and goals (Cao and Ramesh, 2007; Kakar, 2016). As a result, the members of cohesive software development teams display a willingness to participate in group activities while coordinating and interacting effectively (Cao and Ramesh, 2007; Kakar, 2016).

Team cohesion, as a shared psychological state among team members, is a function of team members' interactions (Dey and MP, 2020). When a team has clearly defined roles, clear objectives and goals, and clear priorities and norms, team members can build a shared understanding and establish effective communication. A well-structured team provides its members with more interaction opportunities and improves team cohesion (Dey and MP, 2020). When people collaborate toward achieving shared goals and objectives, they tend to cooperate more and have a stronger sense of community, which ultimately contributes to team cohesion (Wojciechowska-Dzięcielak, 2020; Yang and Lin, 2022). A clear team structure allows a team to facilitate daily interaction and communications, which are vital in fostering team cohesion. Based on these observations, we formulated the following hypotheses:

H1: Team structure positively influences team cohesion in large-scale ASD.

3.3. Knowledge sharing

Knowledge sharing refers to activities in which "employees share their work-related experience, expertise, know-how, and contextual information with other colleagues" (Lin, 2007). As such, knowledge sharing is a way to acquire cognitive resources, such as ideas, information, and knowledge, or transfer those resources from one entity (i.e., employee, team, or organization) to another (Carmeli et al., 2013). To develop capacities and improve competencies, team members should share knowledge, discuss, and reflect on mistakes (Edmondson, 2002a, 2002b). To this end, communication and coordination are crucial, as they allow team members to exchange information and knowledge related to the team, work processes, resources, and context (Edmondson et al., 2007). Prior research suggests that proper knowledge-sharing practices in teams lead to improved decision-making (Davenport et al., 1996), better problem-solving (Kogut and Zander, 1992; Salisbury, 2001), and enhanced creativity (Nonaka and Takeuchi, 1995). Other studies have suggested that sharing information and knowledge with teammates promotes positive psychological states, such as team cohesion (Mesmer-Magnus and DeChurch, 2009; Yang and Lin, 2022). Especially in smaller and less bureaucratic teams (such as ASD teams), which collaborate more efficiently, having a broader range of communication and knowledge-sharing channels enhances team bonding and cohesion (Carmeli et al., 2011; Wojciechowska-Dzięcielak, 2020).

When a team has a clearly defined structure in terms of roles, goals, and priorities, it fosters interactions and communications in the team. The extant ASD literature suggests that practices that facilitate daily interactions and communication between team members help fostering cohesion (Dingsøyr et al., 2016; Peeters et al., 2022), which in turn improves knowledge sharing between project members (Kakar, 2016). For instance, Dingsøyr et al. (2016) report that practices such as daily stand-up meetings, review meetings, or pair programming support cohesion. These practices enable team members to interact frequently and form stronger interpersonal relationships (Peeters et al., 2022), which ultimately enhances team cohesion. Therefore, by improving coordination and collaboration between team members, cohesion enhances knowledge sharing in ASD projects (Kakar, 2016). Thus, we hypothesize:

H2a: Team structure positively contributes to knowledge sharing among team members in large-scale ASD.

H2b: Improved team cohesion positively contributes to knowledge sharing among team members in large-scale ASD.

3.4. Team reflexivity

Team reflexivity refers to a team's ability to engage in regular reflection and learn from past experiences (Peeters et al., 2022). It entails a "dual focus" (LePine et al., 2008), consisting of both reflecting on previous accomplishments and adapting for future actions. Previous research has shown that reflexivity is a key factor influencing team learning (Buvik and Tkalich, 2022; Edmondson, 1999; Schippers et al., 2013). At the team level, reflexivity involves open communication and discussions between team members about "the groups' objectives, strategies, and processes and adapt[ing] them to current or anticipated endogenous or environmental circumstances" (West, 1996, p.559). As such, reflexivity is a collective sense-making process that requires team members to share their viewpoints and ideas openly (Schippers et al., 2008). To this end, team structure, in terms of shared objectives and goals, facilitates communication and information sharing among team members, which in turn enhances team reflexivity (Schippers et al., 2008).

Reflexivity is crucial for software development teams to understand their tasks, dependencies, actions, and interactions. For instance, Scrum as an agile framework helps teams discuss and share an understanding of their project activities and develop a shared mental model, which results in improved performance (Dingsøyr et al., 2016). Daily stand-up

meetings and retrospectives assist a team in reflecting on how to be more efficient, after which the team adjusts and adapts its behavior accordingly. Previous studies on ASD show that reflecting on past decisions enables teams to adjust their processes and become more effective (Babb et al., 2014; Dingsøyr et al., 2016; Peeters et al., 2022). In addition, reflection enables ASD teams to enhance their performance by identifying and prioritizing the most valuable requirements to be addressed (Dingsøyr et al., 2019). In cohesive teams, members are able to build high-quality relationships that lead to more efficient and regular interactions and better information exchange (Tung and Chang, 2011; Yang and Lin, 2022), improving team reflexivity. Further, clearly established goals, objectives, roles, and priorities serve as a reference for open and honest reflection. Thus, we hypothesize:

H3a: Team structure positively affects team reflexivity in large-scale ASD.

H3b: Team cohesion positively affects team reflexivity in large-scale ASD.

3.5. Team learning

Team learning is defined as gaining and sharing skills, knowledge, and information about work through the interaction of members (Argote et al., 2014; Edmondson, 1999). These interactions involve actions such as experimentation, requesting feedback, and discussing errors (Argote et al., 2014). To understand team learning, researchers have examined both internal and external learning behaviors (Argote et al., 2014; Dingsøyr et al., 2016; Wong, 2004). Internal learning pertains to capturing interactions within the team boundary, whereas external learning focuses on interactions beyond the team boundary with individuals and groups. Team learning also involves questioning and reflecting on past actions, mistakes, and their outcomes to identify opportunities for adapting and improving plans (Edmondson, 1999).

In cohesive teams, members tend to share responsibility for team failure (Brawley et al., 1987), produce better software products (Carmel and Sawyer, 1998), and achieve better success rates (Carron et al., 2002) than conflict-ridden teams. Since learning activities encompass discussion of errors and feedback exchange, cohesion is an important team characteristic that can facilitate learning (e.g., Dey and MP, 2020; Edmondson, 1999; Van den Bossche et al., 2006; Yao et al., 2015). Furthermore, team learning can be enhanced when members openly reflect on past actions and mistakes and discuss opportunities for improvement (Bresman and Zellmer-Bruhn, 2013; Carmeli et al., 2009). ASD practices that facilitate frequent communications and social interactions between ASD team members are reported to enhance reflexivity (Peeters et al., 2022), which in turn contributes to learning (Babb et al., 2014; Dingsøyr et al., 2016; Peeters et al., 2022). As such, reflection contributes to team learning, resolves conflicts, and develops trust among members (Lamoreux, 2005). Further, when team members engage in knowledge sharing activities, it improves the team learning outcomes (Buvik and Tkalich, 2022; Ngereja and Hussein, 2021; Peeters et al., 2022). Based on these arguments, we formulated the following hypotheses:

H4a: Team cohesion has a positive impact on team learning in large-scale ASD.

H4b: Team reflexivity has a positive impact on team learning in large-scale ASD.

H4c: Knowledge sharing among team members has a positive impact on team learning in large-scale ASD.

3.6. Control variables

Previous studies suggest that team size has an impact on the effectiveness of agile practices (Peeters et al., 2022). Other studies have reported that team size affects intra-team communication (Kakar, 2016), coordination (Mehta et al., 2022), and knowledge sharing (Carroll et al., 2023; Kakar, 2016; Mehta et al., 2022). While in larger teams the amount of information exchanged is likely to be higher increase with large team sizes, the intra-team interactions and the communication can become challenging (Kakar, 2016). Therefore, based on these observations, we decided to control for team size. We also decided to control for work experience as it is suggested to impact knowledge sharing (Kakar, 2016). A group of studies have reported that the level of familiarity and experience with ASD has an impact on proper adoption of ASD methods (Dikert et al., 2016). Other studies have shown that ASD team members with diverse experiences and expertise help with knowledge sharing and learning (Kakar, 2016; Lee and Xia, 2010). Team members' experience (in terms of tenure) has been reported to influence team learning (Bresman and Zellmer-Bruhn, 2013; Edmondson, 1999).

4. Methodology

4.1. Data collection and sample

To test our theoretical model, we designed an online survey that was divided into three sections. In the first section, the participants were provided with information about the study, its objectives, and its expected benefits for the companies. The second section consisted of 20 questions related to five latent constructs, each rated on a seven-point Likert scale ranging from 1 (strongly disagree) to 7 (strongly agree). We relied on tested scales used in the previous studies to operationalize our constructs (see Table A.1 in Appendix A). To measure team structure, four questions were adopted from Bresman and Zellmer-Bruhn (2013). We also adopted four questions from Kakar (2016) to measure team cohesion. As the questions related to team cohesion are negative statements, we reverse-coded the responses to reflect team cohesion on a positive scale. Knowledge sharing was measured using three items suggested by Kakar (2016). To measure team reflexivity, we adopted five questions from Schippers et al. (2013). Finally, to measure team learning, we used four questions from Edmondson (1999). When needed, we modified the questions to suit the context of the current study.

The last section of the questionnaire included questions related to control variables, including team size and respondents' experience, and demographic questions, including large-scale ASD methods and practices used in projects. Once the responses were received, we operationalized these variables as binary variables. We categorized team size into a common ASD team size of 6–10 members per team (Alqudah and Razali, 2016; Kalenda et al., 2018) and other teams smaller than 5 or larger than 10. Respondent's experience was categorized into less than 4 years and 4 or more years.

Since the items used in the survey have been validated in previous research and proven to have acceptable reliability, we did not conduct a pilot test prior to data collection. However, to ensure that the questions are interpreted correctly, prior to conducting the survey, we tested it with 15 experts from research and industry to assess the structure and content validity of the items used in the study as well as the clarity of the questionnaire. The respondents include five researchers who hold a PhD in software engineering and had more than seven years of experience in teaching ASD and a track record of working on ASD projects with industrial partners. In addition, 10 respondents were software professionals working in large-scale ASD projects in Sweden. This enabled us to evaluate the appropriateness of our survey instrument from a combination of academic and practical perspectives. Based on the feedback received from these experts, we made changes to the wording of the questions or added extra options to our survey questions. For instance, instead of using a long list of titles and roles, we decided to group similar roles (e.g. Scrum Master, Team Coach, etc. were grouped as Team Lead).

Considering the context of the study, we used purposive sampling to recruit respondents who worked in large-scale ASD projects in five Swedish software and systems development companies that operate globally (see Table A.2 in Appendix A). Purposive sampling is useful for selecting respondents that meet certain criteria (e.g. software professionals working in large-scale ASD) which yields appropriate and

useful data for addressing the aims and objectives of the research. We selected these projects because they comprised over 50 software professionals, working in more than five teams, that collaboratively develop and maintain a common product. Therefore, the projects fulfilled the criteria of large-scale ASD projects suggested in the literature (Dikert et al., 2016; e.g., Dingsøyr et al., 2023). Representatives from each company helped distribute the surveys among software professionals participating in these projects. The data collection period lasted three weeks for each large-scale ASD project during Autumn 2022.

4.2. Data analysis

To analyze the survey responses, we opted for a two-step approach using CFA and SEM following guidelines suggested by (Jackson et al., 2009). SEM is a strong multivariate statistical tool that combines path analysis, factor analysis, and simultaneous equation models (Rosseel, 2012) and is suited for describing complex relationships between variables (Weston and Gore Jr, 2006). While the SEM model can estimate the factors, the measurement model, and the path model, it is still important to assess whether the measurement items and survey questionnaire measure the underlying construct as intended. We used covariance-based CFA to assess our measurement model. The choice of measurement model and subsequent assessment method depends on the nature of the theoretical construct. In our model, we posit that the constructs are latent factors reflected in the responses to the survey questions. That is to say, variance in the measurement items is due to the underlying common factors—changes in the latent variables are reflected in the changes in the observable measuring indicators (Diamantopoulos and Siguaw, 2006). Furthermore, because of their reflective nature, we expect the indicators measuring the same latent constructs to correlate with each other and to be interchangeable without affecting the interpretation of latent constructs; we also expect that the removal of an indicator will not change the meaning of the latent constructs (Jarvis et al., 2003). Thus, we first conducted a CFA to ensure that the measurement model fit the data reasonably and that our proposed measurement model exhibited adequate reliability, consistency, and convergent and discriminant validity.

In the next step, we conducted SEM to evaluate the proposed hypothesis. Our analysis was carried out using the R statistical programming language[1] through the RStudio interface.[2] We used the R package Lavaan (Rosseel, 2012), the implementation of SEM and related methods, and SemTools (Jorgensen et al., 2016) to conduct our analysis. In the analysis phases, for both the measurement model and the SEM analysis, we used a maximum likelihood estimator.

5. Results

We received 161 complete survey responses, however, after quality control we excluded two responses leaving us with a sample size of $N = 159$. In particular, we realized that two respondents have stated they are not familiar with ASD at all. On further inspection, we found that these two respondents were relatively novice software professionals with less than one year of experience. Thus, we decided to exclude their responses from the dataset.

In terms of position, the sample is evenly distributed covering various positions, including team lead, architects, developers and testers, business analysts, and others (i.e., product owner, project manager, UX designer, and UI leader). Notably, all the participants demonstrated familiarity with ASD methods, with only 3 % of respondents stating being not so familiar with ASD. The participants operated within diverse team environments, with over 65 % reporting being part of a team comprising 6 to 10 individuals. Additionally, 21 % of the respondents indicated that their team sizes ranged from 11 to 15 individuals. Conversely, 11 % of the participants mentioned being part of a team consisting of 1 to 5 members. In terms of gender distribution, most of the respondents identified as male (62 %). Female participants constituted 34 % of the respondents, while only 3 % preferred not to disclose their gender. Table 1 shows the demographic information about the respondents.

While SAFe ($n = 93$) and LeSS ($n = 66$) were the primary methods adopted in their projects, some respondents reported using other methods such as Extreme Programming ($n = 33$), Scrumban ($n = 12$), Lean Software Development ($n = 6$), and Kanban ($n = 5$) too. In addition, the respondents reported using various agile practices for coordination, knowledge sharing, and reflection. Daily stand-up was almost universal, with 154 out of 159 reporting using it on their team. Practices such as retrospectives ($n = 140$), sprint reviews ($n = 109$), definition of done ($n = 90$), backlog grooming ($n = 89$), tackboard ($n = 67$), and pair programming ($n = 72$) were also frequently reported. It must be noted that the projects followed iterations with various lengths from one to four weeks, while three-week sprints were common in the project following SAFe and two-week sprints in projects following LeSS.

5.1. Measurement model

Earlier, we operationalized our five latent constructs with a set of related questions. The objective of CFA is to evaluate whether the measuring questions can represent the underlying latent constructs, as proposed in the theoretical model. In covariance-based CFA, we assume that the measurement items related to the same latent constructs share variance (i.e., the shared variance is due to the same underlying latent construct). To estimate our measurement model using a covariance-based CFA model and the covariance matrix as input data. In the first round of analysis, we found that one item in team structure (TS3), one item in team cohesion (TC1), one item in team reflexivity (TR2), one item in knowledge sharing (KS3), and one item in team learning (TL1) had weaker factor loadings and ultimately, Average Variance Extracted (AVE) well below the desired cut-off. As our constructs are reflective in nature, dropping these items does not change the meaning of the construct and can potentially improve the convergent validity of the construct (Russo and Stol, 2021). Thus, following the guidelines by (Russo and Stol, 2021; Sarstedt et al., 2021), we dropped the items with weak loading from our measurement model except for KS3. While KS1 and KS2 focus on voluntary knowledge sharing and cooperation between individuals, KS3 is related to the ability of individuals to acquire knowledge from external sources (see Appendix A), representing a

Table 1
Demographic information of the respondents ($N = 159$).

	Freq.	%		Freq.	%
Education			**Familiarity with ASD**		
Vocational Training	67	42	Extremely Familiar	19	12
Bachelor's Degree	38	24	Very Familiar	82	52
Master's Degree	53	33	Familiar	53	33
Ph.D.	1	1	Not so familiar	5	3
Experience			**Roles**		
Less than 1	27	17	Team Lead	37	23
1–2 years	4	3	Architect	11	7
2–3 Years	12	8	Developer & Tester	65	41
3–4 years	11	7	Business Analyst	18	11
4+ years	105	66	Technical Expert	5	3
Team Size			Others	23	15
1–5 People	18	11	**Gender**		
6–10 people	104	65	Female	55	35
11–15 people	34	21	Male	99	62
20 or more people	3	2	Prefer not to say	5	3

[1] R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

[2] RStudio Team (2020). RStudio: Integrated Development for R. RStudio, PBC, Boston, MA. http://www.rstudio.com/.

different dimension of knowledge sharing. Further, despite a lower factor loading, knowledge sharing still has AVE greater than 0.50, implying satisfactory convergent validity. Therefore, following the suggestions by (Sarstedt et al., 2021), we decided to retain the item. After dropping items with weak loadings, except for KS3, we re-estimated the CFA model.

In the first step, we assessed the fit between our proposed theoretical model and data using various goodness-of-fit measures. Our model showed a significant p-value for the chi-square test statistic, 0.07 (test statistic: 131.38, degrees of freedom: 94), rejecting a perfect fit between the data and the proposed model. While the chi-square test is considered a lone test statistic in evaluating measurement models, it alone can neither guarantee a good model fit nor the results should be discarded based on a single test, as it is still worth considering that the chi-square test is subjected to model complexity (favoring complex over the parsimonious model), multivariate normality and other model assumptions, and sample size (Alavi et al., 2020; Schermelleh-Engel et al., 2003). Nevertheless, as good practice, we inspected the chi-square test statistic and explored other additional model fit indices. In particular, we evaluated multiple global and local fit indices, such as the comparative fit index (CFI), Tucker–Lewis index (TLI), root mean square error of approximation (RMSEA), and standardized root mean square residual (SRMR), which are widely used to evaluate the goodness of model fit. With a CFI of 0.96 and TLI of 0.95, our model meets the criteria of a good fit between the data and the model as recommended in the literature (for details, see Hu and Bentler, 1999; Schermelleh-Engel et al., 2003; Shi et al., 2019).

The CFI and TLI indices show that our model shows a desirable fit with the data. The RMSEA is an absolute fit index that indicates how well the hypothesized model fits the data compared to a model that perfectly fits the data (Xia and Yang, 2019). As the fitted residuals are dependent on the scale and the value of the RMSEA is unstandardized, it is difficult to interpret these values (Shi et al., 2018). As an alternative, the SRMR, an absolute model fit index (like the RMSEA), is also based on residuals but is standardized to overcome the issues of scale dependence. Therefore, it can be used as a test statistic for a model fit in addition to the RMSEA. Our model exhibits RMSEA of 0.05 (below 0.05 recommended but below 0.8 is acceptable) and SRMR of 0.65 (below 0.05 recommended but below 0.1 is acceptable), respectively. Additionally, we rejected the hypothesis that the RMSEA of the proposed model was greater than 0.08 at a 95 % confidence interval with a p-value of 0.04, demonstrating a fit between the data and the model. These results show that our proposed measurement model exhibits adequate goodness of fit and conclude that the fit between data and the empirical model allows us to investigate the measurement model further.

Table 2 presents the descriptive summary and various measures of the consistency, reliability, and validity of our proposed measurement model. All the factor loadings are statistically significant and within the desirable range of 0.5 to 0.7, except for one indicator. As explained earlier, the third indicator in knowledge sharing (KS3) representing a separate dimension has a factor loading of 0.47. As per Streiner (2003), our model still exhibits adequate fit (i.e., all constructs are internally consistent with Cronbach's alpha above 0.7) and reliability (i.e., all composite reliability scores above 0.7), we retain the indicator. Next, we assessed both the convergent and discriminant validity of the construct. The average variance extracted (i.e., the proportion of the variance in latent constructs explained by the measuring items) is one of the most frequently used measures of convergent validity. For convergent validity, a rule of thumb is that measuring items should explain at least 50 % of the variance in constructs, with an Average Variance Extracted (AVE) above 0.5. Despite a lower factor loading of the indicator KS3, knowledge sharing exhibited convergent validity with an AVE of 0.528. However, team cohesion, team learning, and team reflexivity did not exceed the recommended cutoff of 0.5, even though the values are not significantly lower than the cutoff value. Table B.1 and B.2.

The value of AVE is calculated from the standardized factor loading, the average of the squared standardized factor loadings, and any factor loadings below 0.7, which are highly likely to be reflected on the AVE and convergent validity. Across these three constructs, only one item has factor loadings above 0.7 and thus has contributed to the AVE below 0.5. Nevertheless, Fornell and Larcker (1981) suggest that the research may claim convergent validity if AVE is significantly greater than 0.5, which is a conservative measure. They also suggest that the researcher can also rely only on the composite reliability measure even if more than half of the variance is due to measurement error, implying AVE lower than 0.5. Given the AVEs are not far from the cutoff value of 0.5 and the constructs have adequate composite reliability, all above (0.7), we still contend that the constructs exhibit convergent validity. Finally, we assessed the discriminant validity based on the factor correlation test as suggested by Rönkkö and Cho (2022). All the factor correlations were below 0.8, indicating the presence of discriminant validity (see Table B.3, Appendix B).

5.2. Structural model

Once we established the measurement model, we estimated the structural equation model to test our hypothesis. Our results show that team structure improves team cohesion, and thus, the first hypothesis is supported. Contrary to our expectations, hypotheses 2a (team structure and knowledge sharing) and 2b (team cohesion and knowledge sharing) were not supported. While the impact of team structure and cohesion on knowledge sharing was positive, it was not statistically significant. Considering the complex nature of large-scale ASD, we speculate that the coordination and communication practices (i.e., attributes of

Table 2
Mean, variance, and standardized factor loadings of indicators estimated based on the CFA model.

Construct	Item	Mean	Variance	Item Loading	Std. Error	P-value	(CA)	(CR)	(AVE)
Team Learning (TL)	TL2	5.41	1.6	0.686	0.076	0.00	0.745	0.738	0.49
	TL3	5.6	1.81	0.752	0.074	0.00			
	TL4	5.99	0.92	0.659	0.077	0.00			
Team Reflexivity (TR)	TR1	4.99	2.18	0.672	0.077	0.00	0.795	0.791	0.491
	TR3	4.91	2.24	0.691	0.076	0.00			
	TR4	5.18	2.04	0.754	0.074	0.00			
	TR5	5.77	1.29	0.681	0.076	0.00			
Knowledge Sharing (KS)	KS1	6.43	0.68	0.800	0.078	0.00	0.733	0.769	0.528
	KS2	6.21	0.82	0.847	0.077	0.00			
	KS3	5.7	1.88	0.475	0.082	0.00			
Team Cohesion (TC)	TC2	5.72	1.834	0.683	0.079	0.00	0.74	0.751	0.488
	TC3	5.91	1.815	0.691	0.079	0.00			
	TC4	5.74	2.55	0.721	0.078	0.00			
Team Structure (TS)	TS1	5.56	2.31	0.868	0.066	0.00	0.835	0.857	0.667
	TS2	5.62	2.03	0.945	0.064	0.00			
	TS4	5.78	1.31	0.595	0.074	0.00			

structure) in large-scale projects can have an impact on knowledge sharing among team members. While team members might have a good understanding of their goals and objectives (i.e., the attributes of team structure) and bond with their teammates through regular intra-team interactions (i.e., the attributes of team cohesion), issues related to inter-team interactions and collaboration in large-scale ASD can diminish these benefits and make knowledge sharing challenging. However, all hypotheses related to team reflexivity (3a and 3b) and team learning (4a, 4b, and 4c) were supported. All the independent variables—team cohesion, knowledge sharing, and team reflexivity—have a positive impact on team learning, with team cohesion being the strongest, followed by knowledge sharing and team reflexivity. Table 3 presents a summary of the hypothesis testing results based on the regression model estimation of the SEM. Table 4.

Fig. 2 shows our final model with standardized coefficients. The two constructs, team structure and knowledge sharing, explain team cohesion, with an r-square value of 0.303—meaning that the two constructs explain 30 % of the variance in team cohesion, even if the impact of knowledge sharing is not statistically significant. Similarly, team cohesion and reflexivity explain 38.5 % of the variance in the team reflexivity constructs. Finally, three constructs—knowledge sharing, team cohesion, and team reflexivity—explain team learning with an r-squared of 0.775, showing strong explanatory power. Controlling for team size and respondent's experience revealed that none of these control variables has a significant effect on the hypothesized relationships, except for experience that negatively affected team learning (see Table B.4 in Appendix B). An explanation for this observation can be that more experienced individuals are often the ones that share information and knowledge with others, and therefore, they might think that they do not learn or gain enough new knowledge from others. Another explanation can be that as software professionals become more experienced, they become more critical towards challenges or shortcomings related to learning activities in their projects.

We operationalized our control variables as binary variables. For instance, we coded team size 6–10 as one and other team sizes as zero. Similarly, we coded software developers' experience, where experience over four years was coded as one and less than four as zero. All the effects of control variables were insignificant, except the effect of experience on learning, which was negative (−0.15). To ensure the robustness of our analysis, we varied team size to a) 1 if larger than 10 else zero and b) 1 if between 1–5 else zero. Similarly, we varied software developer's experience to a) 1 if more than 3 years else zero and b) 1 if less than three years else zero. We evaluated two additional models, and the main impact of variables did not change significantly, and the main results still held. The impact of team size remained insignificant throughout these models. However, the impact of software developer experience turned insignificant when operationalized at the threshold of three years, casting the need for further investigation. Fig. B.1

Table 3
Summary of the hypothesized relationships and standardized coefficients.

Hypothesis	Estimate	Standard Error	P-value	Supported
H1: Team Structure → Team Cohesion (+)	0.608	0.137	0.000	Yes
H2a: Team Structure → Knowledge Sharing (+)	0.158	0.130	0.212	No
H2b: Team Cohesion → Knowledge Sharing (+)	0.097	0.112	0.488	No
H3a: Team Structure → Team Reflexivity (+)	0.287	0.143	0.015	Yes
H3b: Team Cohesion → Team Reflexivity (+)	0.325	0.129	0.012	Yes
H4a: Team Cohesion → Team Learning (+)	0.403	0.212	0.001	Yes
H4b: Team Reflexivity → Team Learning (+)	0.397	0.217	0.002	Yes
H4c: Knowledge Sharing → Team Learning (+)	0.400	0.228	0.000	Yes

6. Discussion

Learning in organizational contexts has received significant attention in the social sciences and organizational studies disciplines. However, there is a distinct gap in the current literature when it comes to learning in software development, especially in the context of large-scale ASD. Specifically, there is a dearth of studies that focus on the unique dynamics and characteristics of learning environments in large-scale ASD and their impact on software professionals' engagement in learning behaviors. As a first step in addressing this gap, we examined five large-scale ASD projects in Sweden. Our findings have several important implications for both software development research and practice in general, especially for the body of knowledge on large-scale ASD. In the following section, we discuss the contributions of this study to software development research and practice.

6.1. Theoretical contributions and practical implications

Our study makes several contributions to the ASD and large-scale ASD literature. Previous studies show that the work environment has an impact on the adoption of ASD methods and their use across organizations and projects (Altuwaijri and Ferrario, 2022; Chow and Cao, 2008; Strode et al., 2009; Tolfo and Wazlawick, 2008) as well as ASD teams' performance (Ahmad, 2025b; Rožman et al., 2023). Other studies draw on the teamwork quality (TWQ) construct proposed by (Hoegl and Gemuenden, 2001) to examine the impact of teamwork quality on team performance (e.g., see Dingsøyr et al., 2024; Lindsjørn et al., 2016; Poth et al., 2023), mostly in traditional software development or small-scale ASD settings (Lindsjørn et al., 2016), and team learning as one aspect that contributes to project success. This aligns with our model, which considers team structure and cohesion as fundamental components of a learning environment. However, our model also incorporates knowledge sharing and team reflexivity as other factors that play a significant role in facilitating learning in large-scale ASD projects. As such, in this study, we provide a deeper understanding of the characteristics of the environment in large-scale ASD projects that impact software professionals' engagement in learning behavior. More precisely, our results show a strong relationship between knowledge sharing, team cohesion, and reflexivity, while team structure impacts team cohesion and reflexivity. Based on these observations, we argue that successful learning in large-scale ASD is reliant upon a team's ability to share knowledge, reflect upon experiences, and offer constructive feedback. Such interactions allow team members to openly discuss their actions and mistakes (Buvik and Tkalich, 2022; Ngereja and Hussein, 2021; Peeters et al., 2022). However, considering their interpersonal nature, we argue that these activities require a suitable learning environment that encourages individuals to interact, collaborate, and engage in constructive discussions (Decuyper et al., 2010; Rebelo et al., 2019). Considering this, team structure serves as a catalyst for improving team cohesion and social interaction. However, it is important to note that an excessively rigid team structure, when isolated from other factors and solely focused on structural and procedural aspects, can lead to redundant tasks and hinder desirable adaptability (Langholf and Wilkens, 2021).

We also contribute to the large-scale ASD literature by empirically examining the preliminary findings reported by previous qualitative studies. Most studies on learning in large-scale ASD are qualitative in nature (Dingsøyr et al., 2023; Dingsøyr, Moe, et al., 2018), and there is a significant lack of academic research that empirically examines the factors that contribute to learning in this context. Our work provides statistically significant evidence that team structure plays a crucial role in promoting team cohesion and reflexivity. This is in line with studies from other work contexts that have pointed out the impact of team

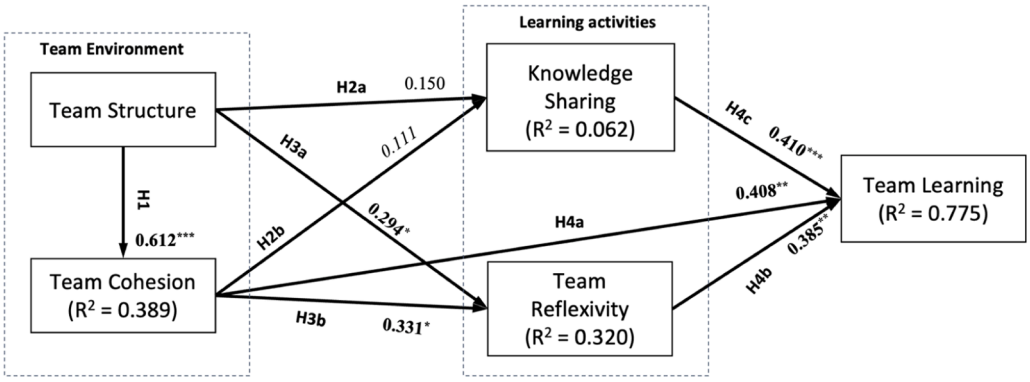

Fig. 2. Final model with standardized coefficients.

structure on team cohesion (Brawley et al., 1993; Callow et al., 2009; Widmeyer and Ducharme, 1997). Additionally, we empirically demonstrated that team structure has an influence on learning that is not fixed but rather contingent on the specific level of structure being examined. Therefore, drawing on the extant studies it can be argued that a better team structure not only decreases conflicts between team members (Bresman and Zellmer-Bruhn, 2013; Bunderson and Boumgarden, 2010) but also encourages team members to engage in active and open reflection on past actions and mistakes and to discuss learning opportunities (Hennel and Rosenkranz, 2021; Mehta et al., 2022).

Finally, our study contributes to the literature on learning in work environments by examining relationships proposed by previous studies in a new context (i.e., large-scale ASD development), thus establishing their generalizability. Considering the sociotechnical and knowledge-intensive nature of software development, our preliminary findings shed light on how different combinations of factors affect learning in large-scale ASD. While the extant literature on learning in work environments suggests a relationship between team structure and cohesion with knowledge sharing (Mesmer-Magnus and DeChurch, 2009; Wojciechowska-Dzięcielak, 2020; Yang and Lin, 2022), our empirical results do not support these relationships. In fact, this observation provides evidence that learning and its underlying dynamics in large-scale ASD differ from other work environments, including small-scale ASD projects in which team cohesion is reported to enhance knowledge sharing (Kakar, 2016). One explanation can be that issues related to inter-team interactions and collaboration in large-scale ASD diminish the positive impact of team structure and team cohesion and consequently make knowledge sharing problematic. This is in line with previous studies showing that knowledge sharing is challenging in large-scale projects due to the larger number of teams and team members (Goh et al., 2013; Kalenda et al., 2018) and more complex knowledge dependencies (Lindsjørn et al., 2018; Strode et al., 2022). Another possible explanation is that imposing rigid structures in large-scale ASD, for example through implementing connecting practices and formal inter-team coordination and communication, can reduce the opportunities for informal social interactions between practitioners and consequently make knowledge sharing more challenging. This argument can be supported by previous studies that report the negative impact of rigid and formal structures in ASD (Langholf and Wilkens, 2021) and large-scale ASD (Annosi et al., 2020). Further longitudinal studies are needed to investigate the interrelationship between team structure and cohesion and their long-term impact on the learning environment in large-scale ASD.

Our study also has implications for managers and software companies. Our results highlight the significant impact that the learning environment has on fostering team learning in a broad sense. This suggests that companies need to acknowledge this impact and take the necessary steps to create a suitable learning environment that enhances learning within and across teams. In doing so, firms need to invest in defining goals, clarifying roles, and establishing effective communication and feedback protocols to enable knowledge sharing and reflection. This could be achieved by following large-scale ASD methods or frameworks that help facilitate coordination and communication in projects. For example, both SAFe[3] and LeSS[4] recommend establishing project-level structures such as community of practice and facilitating regular project-level meetings such as team retrospectives that allow for orchestrating tasks and assignments, sharing knowledge, and tracking progress. SAFe also recommends promoting a culture of innovation to enable continuous learning, while LeSS recommends conducting overall retrospectives after team retrospectives to address cross-team issues and foster learning across teams. In addition, by recognizing the benefits of team structure, managers can promote its implementation and provide the necessary resources and support to enable teams to establish clear roles and priorities. Finally, by creating a safe environment and facilitating continuous interaction within and among teams, managers can encourage team members to engage in open communication as well as constructive reflection and questioning, enabling them to learn individually and collectively from their past experiences and identify opportunities for improvement. To this end, managers and team leaders can implement different practices, such as team-building activities, establishing shared goals, and providing collective rewards, to promote team cohesion.

6.2. *Threats to validity, limitations and future research*

While our study provides valuable insights into the dynamics of team learning within large-scale ASD projects, it is important to interpret the findings with caution due to several limitations. Below, we discuss some potential limitations in terms of construct, internal external, and statistical conclusion validity, as suggested by (Wohlin et al., 2012).

Regarding **internal validity,** we utilized validated survey measurement items from the literature. However, since we relied on self-reported data from software professionals to assess the variables, the data can be subject to bias, including social desirability bias and recall bias. We also acknowledge that we did not employ questions for common method bias and attention checks, which are more common in other survey studies. As our respondents were software professionals participating voluntarily, we refrained from including questions for attention checks and common method bias. Therefore, future research could benefit from incorporating objective measures or third-party assessments to validate the findings. We also acknowledge that there could be other factors that affect team learning that have not been accounted for. While we considered some essential factors that impact learning in large-scale ASD, our model primarily investigates the impact of team-level factors on team learning. However, other factors, such as

[3] https://framework.scaledagile.com/
[4] https://less.works/

leadership, psychological safety, and interaction with non-software development stakeholders and other teams, are also reported to affect team learning. Therefore, future research is needed to investigate the impact of additional project-level and organizational-level factors on team learning in large-scale ASD, especially by conducting inter-team and multi-level analyses. Future research can also consider individual and contextual factors that influence the effectiveness of knowledge sharing for team learning. For example, factors such as an individual's attitude toward knowledge sharing, their level of expertise, and the team's overall openness to learning may all have an impact. Furthermore, as our model mainly focused on team learning, we only controlled for the effect of team size and years of experience in software development. However, future research can investigate the robustness of our results when controlling for other technical and non-technical factors like personality traits, agile practices and tools used in a project, frequency of communication, as well as the context of applications such as IoT applications, software development, and platform developments.

Regarding **construct validity**, the main concern is about using the right operational measures in the survey. We relied on tested and widely used measures proposed by the extant literature. Even though the survey questionnaire was tested by 15 experts before data collection, we did not pilot-test the survey with a larger sample, which can be considered a limitation. Nevertheless, after conducting the survey, we assessed the goodness of model fit to ensure that our proposed model fit the data adequately. We conducted a thorough evaluation of the measurement model, and it exhibited an acceptable level of model fit, reliability, and discriminant validity, but the convergent validity can still be a matter of concern. We observed some weaker factor loadings that were subsequently reflected on AVE, suggesting the convergent validity of the construct Team Learning, Team Reflexivity, and Team Cohesion were borderline. This may raise concerns related to the suitability or adaptability of these scales in large-scale agile studies. Future studies should exercise caution in operationalizing these constructs, and we need more studies to scrutinize the validity of these constructs in the context of large-scale software development. In addition, our study was cross-sectional (i.e., performed on a population at one specific point in time), limiting our ability to make causal interpretations based on the collected data. Thus, future longitudinal studies are needed to establish causal relationships between the variables in our model. Future studies can also provide us with a deeper and more nuanced understanding of the antecedents and outcomes of learning in large-scale ASD. This could entail analyzing the mechanisms that facilitate team learning in large-scale ASD projects and examining the effectiveness of various knowledge sharing practices (e.g., communities of practice) and how these contribute to team performance.

Questions regarding the generalizability of the results and the representativeness of our sample can pose a potential threat to **external validity**. Our results are based on data collected from software professionals working in Swedish organizations. Although the survey was distributed across several projects and organizational contexts, the sample might not be representative of the entire population of software professionals in the large-scale ASD context. In addition, the possibility that the results are bound by the cultural backgrounds of the participants cannot be completely ruled out. Therefore, future studies can use our replication package (Upreti et al., 2025) to examine the generalizability of the results across contexts.

Finally, the relationship between the treatment and the outcome can pose a potential threat to **statistical conclusion validity**. To mitigate such threats, in testing our model, we followed guidelines and statistical techniques recommended by (Jackson et al., 2009) which are widely used across scientific disciplines. Before testing the hypothesized relationships, we followed a well-established procedure of validating the measurement model first within the CB-SEM framework. CB-SEM is a classic statistical multi-variate technique widely adopted and used in several domains. It is also the preferred technique when the constructs are reflective in nature, similar to our study.

7. Conclusions

In this study, we drew on well-established theoretical concepts to hypothesize that successful learning in large-scale ASD projects requires a suitable learning environment that empowers software professionals to interact frequently and openly, reflect on their actions, and share their knowledge. We hypothesized that this learning environment is shaped by team structure and team cohesion. The learning environment, in turn, impacts knowledge sharing and team reflexivity (i.e., learning activities), which contribute to learning in large-scale ASD projects. We tested our model with data collected from five large-scale ASD projects. All the hypotheses were supported except for two; contrary to our expectations, while both team structure and team cohesion had a positive impact on knowledge sharing, their impacts were not statistically significant. Therefore, further empirical research is needed to scrutinize these relationships and explain under what conditions team structure and team cohesion affect knowledge sharing. As such, our study expands the body of knowledge on software development, especially in the context of large-scale ASD, and establishes a foundation for future investigations of learning, with the potential to significantly advance our understanding of its antecedents, underlying mechanisms, and outcomes in large-scale ASD projects. In addition, our suggested model provides guidance for IT organizations to facilitate software professionals' engagement in learning activities and behaviors to ultimately raise the likelihood of achieving optimal performance.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used Grammarly to improve the language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

CRediT authorship contribution statement

Muhammad Ovais Ahmad: Writing – original draft, Validation, Methodology, Investigation, Funding acquisition, Conceptualization. **Hadi Ghanbari:** Writing – review & editing, Writing – original draft, Visualization, Validation, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Tomas Gustavsson:** Writing – original draft, Investigation, Data curation, Conceptualization. **Bikesh Raj Upreti:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Muhammad Ovais Ahmad reports financial support was provided by Knowledge Foundation Sweden, through the NODLA project. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was carried out within the NODLA project [grant no. 20200253] funded by Knowledge Foundation Sweden, Helge Ax: son Johnsons Stiftelse and Swedish Foundation for International Cooperation in Research and Higher Education (STINT) [IB2020–8720]. We would like to thank the anonymous survey respondents for their valuable contribution to this research.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.jss.2025.112561.

Appendix A: Additional information about the survey items and sample

Table 4
Measurement Items.

Constructs (Source)	Item Code	Measurement items
Team Learning (Edmondson, 1999)	TL1	People in our team often speak up to test assumptions about issues under discussion.
	TL2	Our team frequently seeks new information that leads us to make important changes.
	TL3	Our team members go out and get all the information they possibly can from others (e.g. customers, other teams, Internet etc.).
	TL4	People in our team often teach each other new skills.
Team Reflexivity (Schippers et al., 2016)	TR1	The methods used by our team to get the job done are often discussed.
	TR2	Our team often reviews its objectives.
	TR3	We regularly discuss whether our team is working effectively together.
	TR4	Our team often reviews the ways we get the job done.
	TR5	We regularly take time to figure out ways to improve our team's work processes.
Team Cohesion (Kakar, 2016)	TC1	Members of our team would rather go out on their own than get together as a group.
	TC2	I am unhappy with the level of our team's desire to excel.
	TC3	Our team does not give me enough opportunities to improve my personal performance
	TC4	I do not like the way our team is managed.
Knowledge Sharing (Kakar, 2016)	KS1	I voluntarily share my know-how, information, and knowledge with other employees.
	KS2	I cooperate or communicate with other employees in teams or groups for sharing information and knowledge.
	KS3	I can freely access documents, information, and knowledge held by other groups within our team.
Team Structure (Bresman and Zellmer-Bruhn, 2013)	TS1	It is clear what each person in our team is supposed to do.
	TS2	Our team members are sure about what they are expected to do.
	TS3	Our team leader makes sure that the team has clear, explicit expectations for its performance.
	TS4	It is clear to our team members what work goals and objectives are most important—we have a good sense of the priorities of our goals.

Table A.2
Large-scale ASD projects from which data was collected.

#	Company description	Project description	Method used
1	Large Swedish bank and insurance company with over 2000 employees working in the Scandinavian countries.	A project with over 100 software professionals, divided in 12 teams of 5 to 10 people.	SAFe
2	Consulting firm with over 19,000 employees working in 40 countries. The firm offers software and hardware development services to clients.	A project with 70 software professionals divided in 8 teams of 5 to 9 people located in the Nordic countries.	LeSS
3	Leading IT and software service provider in the Nordic region with over 24,000 employees worldwide across more than 20 countries.	A project with 90 software professionals divided in 11 teams of 3 to 7 people.	LeSS
4	Large Nordic bank with over 11,000 employees.	A project with 80 software professionals divided in 9 teams of 6 to 15 people.	LeSS
5	Mid-size software company developing services for ERP systems and operates in the Nordic countries.	A project with 60 software professionals divided in 9 teams of 4 to 8 people.	LeSS

Appendix B: Additional information about the CFA and SEM results

Table B.1
Input Covariance Matrix of the observed items for CFA.

Items	TL2	TL3	TL4	TR1	TR3	TR4	TR5	KS1	KS2	KS3	TC3	TC3	TC4	TS1	TS2	TS4
TL2	1.60	0.85	0.61	0.54	0.63	0.78	0.48	0.24	0.24	0.46	0.46	0.65	0.49	0.67	0.78	0.46
TL3	0.85	1.81	0.62	0.66	0.51	0.72	0.45	0.39	0.44	0.66	0.65	0.74	0.65	0.85	0.87	0.59
TL4	0.61	0.62	0.92	0.32	0.22	0.60	0.33	0.26	0.27	0.31	0.39	0.44	0.53	0.27	0.34	0.32
TR1	0.54	0.66	0.32	2.18	1.10	1.07	0.71	0.16	0.17	0.23	0.41	0.42	0.74	0.77	0.77	0.60
TR3	0.63	0.51	0.22	1.10	2.24	1.06	0.92	0.08	0.07	0.10	0.39	0.33	0.51	0.64	0.53	0.52
TR4	0.78	0.72	0.60	1.07	1.06	2.04	0.80	0.09	0.11	0.18	0.44	0.61	0.89	0.68	0.60	0.67
TR5	0.48	0.45	0.33	0.71	0.92	0.80	1.29	0.08	0.06	0.14	0.30	0.34	0.49	0.47	0.51	0.54

(continued on next page)

Table B.1 (*continued*)

Items	TL2	TL3	TL4	TR1	TR3	TR4	TR5	KS1	KS2	KS3	TC3	TC3	TC4	TS1	TS2	TS4
KS1	0.24	0.39	0.26	0.16	0.08	0.09	0.08	0.68	0.51	0.39	0.16	0.14	0.20	0.16	0.20	0.20
KS2	0.24	0.44	0.27	0.17	0.07	0.11	0.06	0.51	0.82	0.50	0.13	0.11	0.05	0.14	0.17	0.11
KS3	0.46	0.66	0.31	0.23	0.10	0.18	0.14	0.39	0.50	1.88	0.47	0.29	0.39	0.36	0.45	0.42
TC3	0.46	0.65	0.39	0.41	0.39	0.44	0.30	0.16	0.13	0.47	1.83	0.86	1.12	0.56	0.74	0.44
TC3	0.65	0.74	0.44	0.42	0.33	0.61	0.34	0.14	0.11	0.29	0.86	1.81	1.02	0.61	0.76	0.51
TC4	0.49	0.65	0.53	0.74	0.51	0.89	0.49	0.20	0.05	0.39	1.12	1.02	2.55	0.91	0.96	0.70
TS1	0.67	0.85	0.27	0.77	0.64	0.68	0.47	0.16	0.14	0.36	0.56	0.61	0.91	2.31	1.79	0.89
TS2	0.78	0.87	0.34	0.77	0.53	0.60	0.51	0.20	0.17	0.45	0.74	0.76	0.96	1.79	2.03	0.89
TS4	0.46	0.59	0.32	0.60	0.52	0.67	0.54	0.20	0.11	0.42	0.44	0.51	0.70	0.89	0.89	1.31

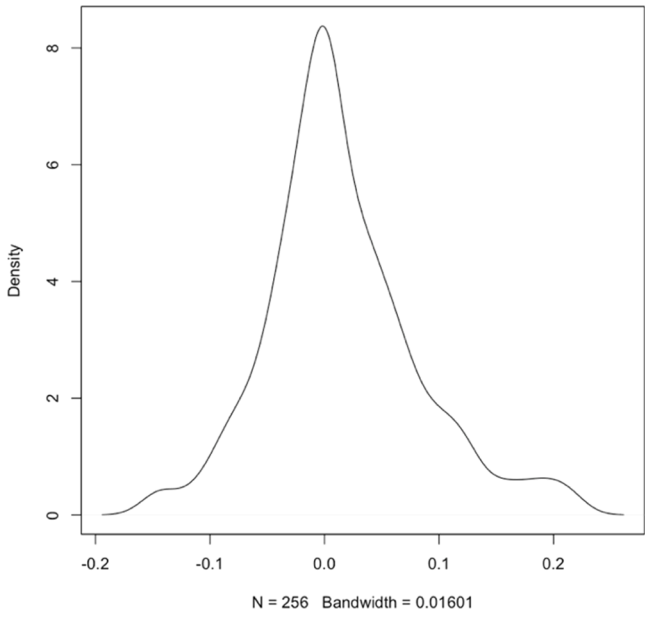

Fig. B.1. Residual Covariance distribution of estimated CFA model.

Table B.2
Reliability and Validity measures for the CFA model.

Constructs/ Criterion	Team Learning (TL)	Team Reflexivity (TR)	Knowledge Sharing (KS)	Team Cohesion (TC)	Team Structure (TS)
Cronbach's Alpha (CA)	0.745	0.795	0.733	0.74	0.835
Average Variance Explained (AVE)	0.49	0.491	0.528	0.488	0.667
Composite Reliability (CR)	0.738	0.791	0.769	0.751	0.857
Maximum Correlation (MC)	0.667	0.647	0.545	0.667	0.597
Square-Root of AVE	0.7	0.701	0.727	0.698	0.816

Table B.3
Test of Discriminant Validity using factor correlation test.

Construct 1	Construct 2	Estimate	CI Lower	CI Upper	Chisq	P-Value
Team Learning	Team Reflexivity	0.647	0.506	0.788	201.502	0.000
Team Learning	Knowledge Sharing	0.545	0.390	0.701	217.458	0.000
Team Learning	Team Cohesion	0.667	0.520	0.813	177.772	0.000
Team Learning	Team Structure	0.584	0.446	0.721	240.301	0.000
Team Reflexivity	Knowledge Sharing	0.151	−0.037	0.340	278.143	0.000
Team Reflexivity	Team Cohesion	0.511	0.347	0.675	206.395	0.000
Team Reflexivity	Team Structure	0.487	0.341	0.632	263.915	0.000
Knowledge Sharing	Team Cohesion	0.200	0.007	0.394	257.320	0.000
Knowledge Sharing	Team Structure	0.209	0.037	0.381	274.926	0.000
Team Cohesion	Team Structure	0.597	0.461	0.733	204.512	0.000

Table B.4
Effect of control variables (as standardized coefficient).

Dependent Variable	Control Variable	Coeff.	Std. Err	P-Value
Team Cohesion	Team Size	−0.118	0.216	0.141
	Developer's Experience	−0.064	0.216	0.425
Knowledge Sharing	Team Size	−0.032	0.190	0.390
	Developer's Experience	−0.074	0.189	0.716
Team Reflexivity	Team Size	0.022	0.206	0.782
	Developer's Experience	0.076	0.204	0.344
Team Learning	Team Size	0.045	0.302	0.510
	Developer's Experience	***−0.150***	***0.324***	***0.039***

Data availability

Anonymised data is shared in our replication package (Upreti et al., 2025).

References

Ahmad, M.O., 2025a. Effective handling of large scale agile secure solution development teams. J. Decis. Syst. 34 (1), 2458875.

Ahmad, M.O., 2025b. Strengthening large-scale agile teams: the interplay of high-quality relationships, psychological safety, and learning from failures. J. Softw. 37 (1), e2759.

Alavi, M., Visentin, D.C., Thapa, D.K., Hunt, G.E., Watson, R., Cleary, M., 2020. Chi-square for model fit in confirmatory factor analysis. J. Adv. Nurs. 76 (9), 2209–2211.

Alqudah, M., Razali, R., 2016. A review of scaling agile methods in large software development. Int. J. Adv. Sci. Eng. Inf. Technol. 6 (6), 828–837.

Altuwaijri, F.S., Ferrario, M.A., 2022. Factors affecting agile adoption: an industry research study of the mobile app sector in Saudi Arabia. J. Syst. Softw. 190, 111347.

Annosi, M.C., Foss, N., Martini, A., 2020. When agile harms learning and innovation:(and what can be done about it). Calif. Manage. Rev. 63 (1), 61–80.

Argote, L., Gruenfeld, D., Naquin, C., 2014. Group learning in organizations. Groups At Work. Psychology Press, pp. 383–426.

Babb, J., Hoda, R., Nørbjerg, J., 2014. Embedding reflection and learning into agile software development. IEEe Softw. 31 (4), 51–57.

Berntzen, M., Hoda, R., Moe, N.B., Stray, V., 2022. A taxonomy of inter-team coordination mechanisms in large-scale agile. IEEE Trans. Softw. Eng. 49 (2), 699–718.

Berntzen, M., Stray, V., Moe, N.B., Hoda, R., 2023. Responding to change over time: a longitudinal case study on changes in coordination mechanisms in large-scale agile. Empir. Softw. Eng. 28 (5), 114.

Brawley, L.R., Carron, A.V., Widmeyer, W.N., 1987. Assessing the cohesion of teams: validity of the group environment questionnaire. J. Sport Exercise Psychol. 9 (3), 275–294.

Brawley, L.R., Carron, A.V., Widmeyer, W.N., 1993. The influence of the group and its cohesiveness on perceptions of group goal-related variables. J. Sport Exercise Psychol. 15 (3), 245–260.

Bresman, H., Zellmer-Bruhn, M., 2013. The structural context of team learning: effects of organizational and team structure on internal and external learning. Organiz. Sci. 24 (4), 1120–1139.

Britto, R., Šmite, D., Damm, L.-O., 2016. Experiences from measuring learning and performance in large-scale distributed software development. In: Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement.

Bunderson, J.S., Boumgarden, P., 2010. Structure and learning in self-managed teams: why "bureaucratic" teams can be better learners. Organiz. Sci. 21 (3), 609–624.

Buvik, M.P., Tkalich, A., 2022. Psychological Safety in Agile Software Development teams: Work Design Antecedents and Performance Consequences. The 55th Hawaii International Conference on System Sciences, Maui, Hawaii.

Callow, N., Smith, M.J., Hardy, L., Arthur, C.A., Hardy, J., 2009. Measurement of transformational leadership and its relationship with team cohesion and performance level. J. Appl. Sport Psychol. 21 (4), 395–412.

Cao, L., Ramesh, B., 2007. Agile software development: Ad hoc practices or sound principles? IT Prof 9 (2), 41–47.

Carmel, E., Sawyer, S., 1998. Packaged software development teams: what makes them different? Inf. Technol. People 11 (1), 7–19.

Carmeli, A., Atwater, L., Levi, A., 2011. How leadership enhances employees' knowledge sharing: the intervening roles of relational and organizational identification. J. Technol. Transf. 36, 257–274.

Carmeli, A., Brueller, D., Dutton, J.E., 2009. Learning behaviours in the workplace: the role of high-quality interpersonal relationships and psychological safety. Syst. Res. Behav.l Sci. 26 (1), 81–98.

Carmeli, A., Gelbard, R., Reiter-Palmon, R., 2013. Leadership, creative problem-solving capacity, and creative performance: the importance of knowledge sharing. Hum Resour Manage 52 (1), 95–121.

Carroll, N., Conboy, K., Wang, X., 2023. From transformation to normalisation: an exploratory study of a large-scale agile transformation. J. Inf. Technol. 38 (3), 267–303.

Carron, A.V., Colman, M.M., Wheeler, J., Stevens, D., 2002. Cohesion and performance in sport: a meta analysis. J. Sport Exercise Psychol. 24 (2), 168–188.

Chow, T., Cao, D.-B., 2008. A survey study of critical success factors in agile software projects. J. Syst. Softw. 81 (6), 961–971.

Conboy, K., 2009. Agility from first principles: reconstructing the concept of agility in information systems development [Article]. Inf. Syst. Res. 20 (3), 329–354. https://doi.org/10.1287/isre.1090.0236.

Conboy, K., Carroll, N., 2019. Implementing large-scale agile frameworks: challenges and recommendations. IEEe Softw. 36 (2), 44–50.

Davenport, T., Jarvenpaa, S., Beers, M., 1996. Improving Knowledge Work Processes. MIT Sloan management review.

Decuyper, S., Dochy, F., Van den Bossche, P., 2010. Grasping the dynamic complexity of team learning: an integrative model for effective team learning in organisations. Educ. Res. Rev. 5 (2), 111–133.

Dey, C., MP, G., 2020. Impact of team design and technical factors on team cohesion. Team Perform. Manage. 26 (7/8), 357–374.

Diamantopoulos, A., Siguaw, J.A., 2006. Formative versus reflective indicators in organizational measure development: a comparison and empirical illustration. British J. Manage. 17 (4), 263–282.

Dikert, K., Paasivaara, M., Lassenius, C., 2016. Challenges and success factors for large-scale agile transformations: a systematic literature review. J. Syst. Softw. 119, 87–108.

Dingsøyr, T., Bjørnson, F.O., Schrof, J., Sporsem, T., 2023. A longitudinal explanatory case study of coordination in a very large development programme: the impact of transitioning from a first-to a second-generation large-scale agile development method. Empir. Softw. Eng. 28 (1), 1–49.

Dingsøyr, T., Fægri, T.E., Dybå, T., Haugset, B., Lindsjørn, Y., 2016. Team performance in software development: research results versus agile principles. IEEe Softw. 33 (4), 106–110.

Dingsøyr, T., Falessi, D., Power, K., 2019. Agile development at scale: the next frontier. IEEe Softw. 36 (2), 30–38.

Dingsøyr, T., Mikalsen, M., Solem, A., Vestues, K., 2018a. Learning in the large-an exploratory study of retrospectives in large-scale agile development. Agile Processes in Software Engineering and Extreme Programming: 19th International Conference, XP 2018. In: . May 21–25, 2018, Proceedings 19.

Dingsøyr, T., Moe, N.B., Fægri, T.E., Seim, E.A., 2018b. Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. Empir. Softw. Eng. 23, 490–520.

Dingsøyr, T., Schneider, P., Bergersen, G.R., Lindsjørn, Y., 2024. Challenges in understanding the relationship between teamwork quality and project success in large-scale agile projects. arXiv preprint. *arXiv:2401.17725*.

Edison, H., Wang, X., Conboy, K., 2022. Comparing methods for large-scale agile software development: a systematic literature review. IEEE Trans. Softw. Eng. 48 (8), 2709–2731.

Edmondson, A., 1999. Psychological safety and learning behavior in work teams. Adm. Sci. Q. 44 (2), 350–383.

Edmondson, A.C., 2002a. The local and variegated nature of learning in organizations: a group-level perspective. Organiz. Sci. 13 (2), 128–146.

Edmondson, A.C., 2002b. Division of Research. Managing the Risk of learning: Psychological safety in Work Teams. Harvard Business School Cambridge, MA.

Edmondson, A.C., Dillon, J.R., Roloff, K.S., 2007. 6 three perspectives on team learning: outcome improvement, task mastery, and group process. Acad. Manage. Ann. 1 (1), 269–314.

Ellis, A.P., Hollenbeck, J.R., Ilgen, D.R., Porter, C.O., West, B.J., Moon, H., 2003. Team learning: collectively connecting the dots. J. Appl. Psychol. 88 (5), 821–835.

Fornell, & Larcker., 1981. Structural equation models with unobservable variables and measurement error: algebra and statistics. Struct. Equ. Models 8.

Garvin, D.A., Edmondson, A.C., Gino, F., 2008. Is yours a learning organization? Harv. Bus. Rev. 86 (3), 109.

Gibson, C., Vermeulen, F., 2003. A healthy divide: subgroups as a stimulus for team learning behavior. Adm. Sci. Q. 48 (2), 202–239.

Gladstein, D.L., 1984. Groups in context: a model of task group effectiveness. Adm. Sci. Q. 499–517.

Goh, J.C.-L., Pan, S.L., Zuo, M., 2013. Developing the agile IS development practices in large-scale IT projects: the trust-mediated organizational controls and IT project team capabilities perspectives. J. Assoc. Inf. Syst. 14 (12), 1.
Gupta, M., George, J.F., Xia, W., 2019. Relationships between IT department culture and agile software development practices: an empirical investigation. Int. J. Inf. Manage 44, 13–24.
Hennel, P., Rosenkranz, C., 2021. Investigating the "socio" in socio-technical development: the case for psychological safety in agile information systems development. Project Manage. J. 52 (1), 11–30.
Hoegl, M., Gemuenden, H.G., 2001. Teamwork quality and the success of innovative projects: a theoretical concept and empirical evidence. Organiz. Sci. 12 (4), 435–449.
Hu, L.t., Bentler, P.M., 1999. Cutoff criteria for fit indexes in covariance structure analysis: conventional criteria versus new alternatives. Struct. Equ. Model. 6 (1), 1–55. https://doi.org/10.1080/10705519909540118.
Jackson, D.L., Gillaspy, Jr, J.A., Purc-Stephenson, R., 2009. Reporting practices in confirmatory factor analysis: an overview and some recommendations. Psychol. Methods 14 (1), 6.
Janz, B.D., Prasarnphanich, P., 2009. Freedom to cooperate: gaining clarity into knowledge integration in information systems development teams. IEEe Trans. Eng. Manage 56 (4), 621–635.
Jarvis, C.B., MacKenzie, S.B., Podsakoff, P.M., 2003. A critical review of construct indicators and measurement model misspecification in marketing and consumer research. J. Consumer Res. 30 (2), 199–218.
Jorgensen, T.D., Pornprasertmanit, S., Schoemann, A., Rosseel, Y., Miller, P., Quick, C., Garnier-Villarreal, M., Selig, J., Boulton, A., Preacher, K., 2016. Package 'semTools. https://cran.r-project.org/web/packages/semTools/semTools.pdf.
Kakar, A.K., 2016. How do team cohesion and psychological safety impact knowledge sharing in software development projects? Knowl. Process Manage. 25 (4), 258–267.
Kalenda, M., Hyna, P., Rossi, B., 2018. Scaling agile in large organizations: practices, challenges, and success factors. J. Softw. 30 (10), e1954.
Kasauli, R., Knauss, E., Horkoff, J., Liebel, G., de Oliveira Neto, F.G, 2021. Requirements engineering challenges and practices in large-scale agile system development. J. Syst. Softw. 172, 110851.
Knaster, R., Leffingwell, D., 2018. SAFe® 4.5 Reference guide: Scaled Agile framework® For Lean Software and Systems Engineering. Addison-Wesley Professional.
Kogut, B., Zander, U., 1992. Knowledge of the firm, combinative capabilities, and the replication of technology. Organiz. Sci. 3 (3), 383–397.
Lamoreux, M., 2005. Improving agile team learning by improving team reflections [agile software development]. In: Agile Development Conference (ADC'05).
Langholf, V., Wilkens, U., 2021. Agile project management, new leadership roles and dynamic capabilities–Insight from a case study analysis. J. Competences Strategy Manage. 11, 1–18.
Larman, C., Vodde, B., 2010. Practices For Scaling Lean & Agile development: large, multisite, and Offshore Product Development with Large-Scale Scrum. Pearson Education.
Lee, G., Xia, W., 2010. Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. MIS Quarterly 34 (1), 87–114.
LePine, J.A., Piccolo, R.F., Jackson, C.L., Mathieu, J.E., Saul, J.R., 2008. A meta-analysis of teamwork processes: tests of a multidimensional model and relationships with team effectiveness criteria. Pers. Psychol. 61 (2), 273–307.
Lin, H., 2007. Effects of extrinsic and intrinsic motivations on employee intentions to share knowledge. J. Inf. Sci. 33 (2), 135–149.
Lindsjørn, Y., Bergersen, G.R., Dingsøyr, T., Sjøberg, D.I., 2018. Teamwork quality and team performance: exploring differences between small and large agile projects. In: Garbajosa, J., Wang, X., Aguiar, A. (Eds.), Teamwork quality and team performance: exploring differences between small and large agile projects. Agile Process. Softw. Eng. Extreme Program. 314, 267–274.
Lindsjørn, Y., Sjøberg, D.I., Dingsøyr, T., Bergersen, G.R., Dybå, T., 2016. Teamwork quality and project success in software development: a survey of agile development teams. J. Syst. Softw. 122, 274–286.
Liu, Z., Stray, V., Sporsem, T.T., 2023. Organizational debt in large-scale hybrid agile software development: a case study on coordination mechanisms. Agile processes in Software engineering and Extreme programming–Workshops: XP 2022 Workshops, Copenhagen, Denmark, June 13–17, 2022, and XP 2023 Workshops, Amsterdam. the Netherlands. June 13–16, 2023, Revised Selected Papers.
Mathieu, J., Maynard, M.T., Rapp, T., Gilson, L., 2008. Team effectiveness 1997-2007: a review of recent advancements and a glimpse into the future. J. Manage 34 (3), 410–476.
Mehta, A., Mehta, N., Bindal, I., 2022. Maximizing integrative learning in software development teams: a systematic review of key drivers and future research agenda. J. Syst. Softw. 190, 1–20.
Mesmer-Magnus, J.R., DeChurch, L.A., 2009. Information sharing and team performance: a meta-analysis. J. Appl. Psychol. 94 (2), 535.
Müller, A., Herbig, B., Petrovic, K., 2009. The explication of implicit team knowledge and its supporting effect on team processes and technical innovations: an action regulation perspective on team reflexivity. Small. Group. Res. 40 (1), 28–51.
Mustaqim, I.Z., Raharjo, T., Trisnawaty, N.W., 2023. Toward unveiling the execution of large-scale agile project in Global software development: a systematic literature review. 2023 Int. Conf. Eng. Emerg. Technol. (ICEET).
Ngereja, B.J., Hussein, B., 2021. An examination of the preconditions of learning to facilitate innovation in digitalization projects: a project team members' perspective. Int. J. Inf. Syst. Project Manage. 9 (2), 23–41.
Nonaka, I., Takeuchi, H., 1995. The Knowledge-Creating company: How Japanese Companies Create the Dynamics of Innovation, 105. Harvard Business Press.
Ozer, M., Vogel, D., 2015. Contextualized relationship between knowledge sharing and performance in software development. J. Manage. Inf. Syst. 32 (2), 134–161.
Paasivaara, M., Behm, B., Lassenius, C., Hallikainen, M., 2018. Large-scale agile transformation at Ericsson: a case study. Empir. Softw. Eng. 23, 2550–2596.
Paasivaara, M., Lassenius, C., 2014. Communities of practice in a large distributed agile software development organization–Case Ericsson. Inf. Softw. Technol. 56 (12), 1556–1577.
Peeters, T., Van De Voorde, K., Paauwe, J., 2022. The effects of working agile on team performance and engagement. Team Perform. Manage. 28 (1/2), 61–78.
Poth, A., Kottke, M., Mahr, T., Riel, A., 2023. Teamwork quality in technology-driven product teams in large-scale agile organizations. J. Softw. 35 (8), e2388.
Putta, A., Paasivaara, M., & Lassenius, C. (2018). Benefits and challenges of adopting the scaled agile framework (SAFe): preliminary results from a multivocal literature review. Product-focused Software process Improvement: 19th International Conference, PROFES 2018, Wolfsburg, Germany, November 28–30, 2018, Proceedings 19.
Qumer, A., Henderson-Sellers, B., 2008. A framework to support the evaluation, adoption and improvement of agile methods in practice. J. Syst. Softw. 81 (11), 1899–1919.
Rebelo, T., Lourenço, P.R., Dimas, I.D., 2019. The journey of team learning since "The Fifth Discipline. Learn. Organiz. 27 (1), 42–53.
Reifer, D.J., Maurer, F., Erdogmus, H., 2003. Scaling agile methods. IEEe Softw. 20 (4), 12–14.
Rönkkö, M., Cho, E., 2022. An updated guideline for assessing discriminant validity. Organ. Res. Methods 25 (1), 6–14.
Rosseel, Y., 2012. lavaan: an R package for structural equation modeling. J. Stat. Softw. 48, 1–36.
Rožman, M., Oreški, D., Tominc, P., 2023. A multidimensional model of the new work environment in the digital age to increase a company's performance and competitiveness. IEEe Access. 11, 26136–26151.
Russo, D., Stol, K.-J., 2021. PLS-SEM for software engineering research: an introduction and survey. ACM Comput. Surv. (CSUR) 54 (4), 1–38.
Salisbury, M., 2001. An example of managing the knowledge creation process for a small work group. Manage. Learn. 32 (3), 305–319.
Sarstedt, M., Ringle, C.M., Hair, J.F., 2021. Partial least squares structural equation modeling. In: Homburg, C., Klarmann, M., Vomberg, A. (Eds.), Handbook of Market Research. Springer, pp. 587–632.
Schermelleh-Engel, K., Moosbrugger, H., Müller, H., 2003. Evaluating the fit of structural equation models: tests of significance and descriptive goodness-of-fit measures. Methods Psychol. Res. Online 8 (2), 23–74.
Schippers, M.C., Den Hartog, D.N., Koopman, P.L., Van Knippenberg, D., 2008. The role of transformational leadership in enhancing team reflexivity. Human Relations 61 (11), 1593–1616.
Schippers, M.C., Homan, A.C., Van Knippenberg, D., 2013. To reflect or not to reflect: prior team performance as a boundary condition of the effects of reflexivity on learning and final team performance. J. Organ. Behav. 34 (1), 6–23.
Shaw, M.E., 1981. Group dynamics: The psychology of Small Group Behavior. McGraw-Hill College.
Shi, D., Lee, T., Maydeu-Olivares, A., 2019. Understanding the model size effect on SEM fit indices. Educ. Psychol. Meas. 79 (2), 310–334.
Shi, D., Maydeu-Olivares, A., DiStefano, C., 2018. The relationship between the standardized root mean square residual and model misspecification in factor analysis models. Multivariate Behav. Res. 53 (5), 676–694.
Strode, D., Dingsøyr, T., Lindsjorn, Y., 2022. A teamwork effectiveness model for agile software development. Empir. Softw. Eng. 27 (2), 56.
Strode, D.E., Huff, S.L., Tretiakov, A., 2009. The impact of organizational culture on agile method use. In: 2009 42nd Hawaii International Conference on System Sciences.
Thorgren, S., Caiman, E., 2019. The role of psychological safety in implementing agile methods across cultures. Res.-Technol. Manage. 62 (2), 31–39.
Tolfo, C., Wazlawick, R.S., 2008. The influence of organizational culture on the adoption of extreme programming. J. Syst. Softw. 81 (11), 1955–1967.
Tung, H.L., Chang, Y.H., 2011. Effects of empowering leadership on performance in management team: mediating effects of knowledge sharing and team cohesion. J. Chin. Human Resourc. Manage. 2 (1), 43–60.
Uludağ, Ö., Philipp, P., Putta, A., Paasivaara, M., Lassenius, C., Matthes, F., 2022. Revealing the state of the art of large-scale agile development research: a systematic mapping study. J. Syst. Softw. 192, 1–43.
Upreti, B.raj, Ghanbari, H., Ahmad, M.O., Gustavsson, T., 2025. It All Starts with Structure: Investigating Learning Dynamics in Large-Scale Agile Software Development - Replication Package. In Journal of Systems and Software. Zenodo.
Van den Bossche, P., Gijselaers, W.H., Segers, M., Kirschner, P.A., 2006. Social and cognitive factors driving teamwork in collaborative learning environments: team learning beliefs and behaviors. Small. Group. Res. 37 (5), 490–521.
van Solingen, R., Berghout, E., Kusters, R., Trienekens, J., 2000. From process improvement to people improvement: enabling learning in software development. Inf. Softw. Technol. 42 (14), 965–971.
West, M., 1996. Reflexivity and Work Group effectiveness: A conceptual Integration. John Wiley & Sons, Ltd.
Weston, R., Gore, Jr, P.A., 2006. A brief guide to structural equation modeling. Couns. Psychol. 34 (5), 719–751.
Widmeyer, W.N., Ducharme, K., 1997. Team building through team goal setting. J. Appl. Sport Psychol. 9 (1), 97–113.
Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., 2012. Experimentation in Software Engineering, 236. Springer.
Wojciechowska-Dzięcielak, P.M., 2020. Knowledge sharing facilitators and barriers in the context of group cohesion—a literature review. Int. J. Inf. Educ. Technol. 10 (1), 31–36.

Wong, S.-S., 2004. Distal and local group learning: performance trade-offs and tensions. Organiz. Sci. 15 (6), 645–656.
Xia, Y., Yang, Y., 2019. RMSEA, CFI, and TLI in structural equation modeling with ordered categorical data: the story they tell depends on the estimation methods. Behav. Res. Methods 51, 409–428.
Yang, H.-H., Lin, Y.-T., 2022. How knowledge sharing and cohesion become keys to a successful graduation project for students from Design College. Sage Open. 12 (3), 21582440221121785.
Yao, C.-Y., Tsai, C.-C., Fang, Y.-C., 2015. Understanding social capital, team learning, members'e-loyalty and knowledge sharing in virtual communities. Total Qual. Manage. Bus. Excellence 26 (5–6), 619–631.