OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 01 - Issue 12 (2024)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

KNOWLEDGE SHARING CHALLENGES AND ITS SOLUTIONS IN LARGE-SCALE AGILE SOFTWARE DEVELOPMENT

Nasir Rashid
Department of Computer Science and IT, University of Malakand Pakistan Software Engineering Research Group, University of Malakand
nrswat744@gmail.com

Badre Alam
Department of Computer Science and IT, University of Malakand Pakistan Software Engineering Research Group, University of Malakand
badrealamuom@gmail.com

Muhammad Salam
Department of Computer Science and IT, University of Malakand Pakistan Software Engineering Research Group, University of Malakand
msmdir@gmail.com

Shah Khalid
Department of Computer Science and IT, University of Malakand Pakistan Software Engineering Research Group, University of Malakand
khaliddir@gmail.com

Haseena Noureen
Department of Computer Science and IT, University of Malakand Pakistan Software Engineering Research Group, University of Malakand
sohny105@yahoo.com

Article Info

ABSTRACT

CONTEXT: *Software development organizations extensively adapt Agile methods due to the promising and appealing benefits and values it adds to the software products and overall software development process. Large-scale agile refers to the process of developing and delivering enterprise systems and software with a significant number of groups. Several studies have been carried out to determine the challenges, still, the research area is in its infancy, and more research is needed to uncover the issues and practices that large-scale agile development is facing.*

OBJECTIVE: *The main purpose of this study is to explore the challenges in knowledge sharing in large-scale agile software development (LSASD), and its relevant solutions/practices to address the identified challenges, and further validate it from software practitioners globally.*

METHOD: *We conducted a Systematic Literature Review (SLR) to identify the challenges that LSASD faces. A total of 48 papers provided data for the identification of challenges and practices. Further, we conducted an online questionnaire survey to validate the SLR results. We received 52 responses from relevant practitioners from various countries.*

RESULTS: *To answer the first research question, we used SLR which resulted in 9 challenges along with the relevant 39 practices. In the second phase, we validated the identified challenges and practices through an online questionnaire survey.*

CONCLUSION: *Large-scale agile software development is desperately needed today. Our efforts will assist software vendor organizations in developing systems while keeping the LSASD's pinpointed issues in mind. These practices will assist the practitioners in overcoming the challenges in knowledge sharing at large-scale agile software development.*

Keywords: *Agile Software Development, Large-Scale Agile Software Development, Knowledge Sharing challenges, systematic literature review.*

1. Introduction

Agile software development is a time-limited, iterative method that produces software incrementally from the beginning of a software project rather than attempting to deliver it all at once. It works by decomposing the projects into user levels, which are small chunks of functionality that are prioritized and delivered in 2-4 weeks iterations [1].

Agile software development is a collection of iterative software development approaches in which requirements and solutions arise from cooperation among self-organizing cross-functional teams. Iteration and frequent feedback it offers are crucial to agile processes, which allow a software system to be refined and delivered over time [2].

The Agile Adoption process is a systematic and repeatable method for guiding and assisting agile adoption activities. It will help the agile community meet the increased demand from businesses looking to use agile techniques [2].

Continuous communication and planned software delivery, continuous software code integration, better project quality and efficiency, minimum documentation, and early project completion, expert client input are just a few of the advantages of using agile methodologies in distributed software development. It gives a conceptual framework for tackling any co-located or globally dispersed software project. Agile approaches, in contrast to traditional software process development, aim to decrease hazards and increase software production by producing software in quick duplications. Agile methods focus on the expertise of individual developers rather than structured processes and a large quantity of documentation. As a result, agile approaches strive to avoid introducing overbearing processes that provide little value to the software output [3]. One of the key advantages is that it allows the customer to submit input and make changes throughout the development phase. When compared to traditional procedures, the enhanced and transparent collaboration with the client results in outputs that better meet their specifications, saving time and cost. Agile software development is iterative and incremental for modifications while the project is still in progress. Because of these features, agile approaches have gained considerable popularity. The customer can submit input during the projects so that adjustments can be made if necessary [3].

Scrum is an agile software development paradigm for managing product development that is iterative and incremental [1]. Extreme programming (XP) is a software development process that aims to improve product quality and flexibility in response to changing client needs. It encourages frequent "releases" in short development cycles as a type of agile software development [1]. Kanban is a prominent agile method for organizing knowledge work that emphasizes just-in-time shipping but avoids overburdening team members [1]. The dynamic systems development method (DSDM) is a way of creating software used for agile project delivery. The main purpose of DSDM is to provide discipline to the rapid application development (RAD) process. DSDM is a technique of software development that is iterative and incremental and contains Agile characteristics such as ongoing user/customer interaction [1].

Due to appealing and practical qualities such as elasticity, awareness, and team encouragement, Agile development methodologies are progressively being employed by large-scale development organizations. Currently, enormous software-intensive firms are using agile methodologies, and there are initiatives to scale agile approaches. Though agile methodologies were originally designed for small development teams, when they were used to agile software development at large-scale, they encountered several difficulties. These include significant problems due to sophisticated technological requirements. These needs can be seen in a variety of ways, including requirements between development activities, requirements between software elements, and requirements across

teams and team members. It demonstrates how to reduces technical needs that have a negative impact on team execution and activities, as well as how to facilitate technical requisite interaction and management amongst teams [4].

Software development is frequently defined as creative activity in which there may be no one best answer and progress toward completion is difficult to predict. large-scale projects involving several teams. Technical and organizational complexity are common in large programs. There are a lot of stakeholders, a lot of participants, a lot of needs, lines of software code, and typically a lot of interdependencies across responsibilities, as well as teams that rely on other teams [5].

In today's research and practice, agile development at large-scale is gaining a lot of attraction. In the context of implementing agile at a large-scale, two main levels should be considered: the individual level, such as a project or team, and the organizational level. The majority of the literature on large-scale agile deployment focuses on individuals or isn't particularly precise [6].

Agile at large-scale is a collaborative effort in which teams must communicate in order to produce software. Hosting Scrum of-scrum meetings to coordinate the work of the many teams is a regular technique to manage large agile projects with several teams. When there are too many people with divergent interests, meetings become inefficient and interminable. Hosting several meetings with teams with shared aims might be a more efficient solution [7].

Motivation

Agile methods are emerging as best practices for software development across the globe. Recently, many software development organizations have started to adopt ASDM methodologies, which are based on a very different philosophy than traditional methods. Therefore, they require different way of thinking for both the team and the management. To gain the benefits of agile methods in at large-scale developmental teams, a complete systematic review is required. Software industries have now realized the competitive advantages of the integrated approach for producing high quality software with accelerated delivery, minimal cost, user satisfaction and flexibility to manage the requirements in the development process. Although agile methods can be adopted for small-scale developmental teams, it has got very little attention from the research community in agile transformation. To bridge the gap and to assist the software project manager in the adoption of agile at large-scale, this research will help the organization in adopting agile at large-scale and will minimize the risks that encounter during the knowledge sharing [3].

Research Questions

The following research issues will be addressed by this study to achieve our goal.

Q1. What are the challenges, in knowledge sharing, as reported in the literature, in implementing agile methods at large-scale?
Q2. What are the challenges, in knowledge sharing, as identified in the real-world practice, in adopting agile methods at large-scale?
Q3. What are the solution/practices, for the identified challenges in knowledge, as reported in the literature, in adopting agile methods at large-scale?
Q4. What are the solution/practices, for the identified challenges in knowledge sharing, in the real world practice, in adopting agile methods at large-scale?
Q5. Is there any difference among the identified challenges in both the data sets (SLR and questionnaire survey)?
RQ6. Do the identified knowledge sharing challenges show any significant variation from one continent to another continent?

1. Background

Agile methods are emerging as best practices for software development across the globe for increased users' satisfaction and accelerated delivery of software. Recently agile methods

have been focused by researchers for the development of software in large-scale. The agile manifesto at large-scale represents a quite innovative work in cleaving and outspreading the critique of formalized software processes over the previous decade and is well acknowledged by practitioners and academicians [2].

Most of the researchers may not have the practical experience of managing the agile software development at a large scale. Adopting Agile methods in larger projects and teams, is difficult as compared to smaller ones -which is the first choice- larger ones will need more coordination. LSAD teams and projects require the involvement of other organizational units like marketing, Human Resources (HR), and product management. In spite of these obvious problems related to LSAD teams and projects, there is an increasing tendency towards adopting them [3].

From the perspective of our interviewees, knowledge sharing is vital to enable good communication and coordination. If knowledge is not properly circulated, communicating technical dependencies will suffer, as indicated by some of the problems raised by the interviewees. Another problem related to knowledge sharing occurs when team members do not understand, ignore, or forget what was discussed in a meeting: “During development some people forget easily what was agreed upon in scrum meetings. Then, they are not able to work accordingly.” From the perspective of our interviewees, it is clear that such problems with knowledge sharing create a major challenge for communicating technical dependencies [4].

We return to our original research topic of knowledge sharing in large-scale agile. Can we, based on these two cases extend our knowledge of this topic? While agile at large scale for sharing knowledge and stresses the importance of continuous improvement, we have seen evidence that the picture is not always so clear on what to do, and different approaches can be taken based on the project type. In a temporary project structure, supporting unofficial meeting arenas was enough, while for learning and gave them decision making authority and responsibilities for continuous process improvement. We have also seen that the need for continuous process improvement might differ from the different projects, where a temporary project structure might optimize on the team and inter-team levels and keep the overarching structures in place. How this affects the development process and whether the process could have been optimized further remains an open question. We have also seen the importance of establishing multiple arenas for knowledge sharing in the beginning of projects in order to accelerate learning across organizational and project boundaries [19].

2. Research Methodology

Our research is divided in two steps. The first step was to conduct a Systematic Literature Review (SLR). We perform a comprehensive study of the literature to determine the challenges of knowledge sharing in Large-Scale Agile Software Development. We also dig out each challenge's practices from literature. The second step is to we have executed a QS conducted real-time. The aim of the QS was to justify the findings of systematic literature review, and to push outside more other than the identified challenges/practices. The research strategy adopted for this research is illustrated in Figure 1.

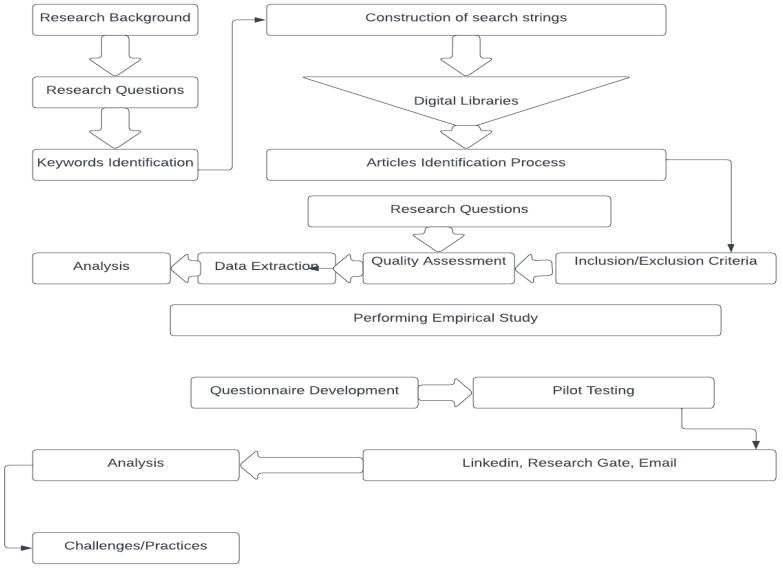

Figure 1: Proposed Study Method

The purpose of this diagram is to show our work for viewers which the author has done. First of all we have constructed search string, then we search many search libraries for the identification of research papers according to our search string. After that we perform inclusion/exclusion criteria, quality assessment, data extraction and then analysis. After obtaining the results, we perform create a questionnaire and share it in linked-in and through email to software developers and experts of software engineering. After that we have responses received from many countries.

3.1 Data Collection

For data collection, we have used SLR and survey.

3.1.1 Systematic Literature Review

In comparison to a traditional literature review, the SLR procedure was chosen for this study because it is more thorough, less biased, and transparent. It has been a trending methodology used in a number of published paper. And the number of papers released each year is growing; below is a list of some papers published in 2021. SLR method have three major phases [8].

- Planning the review/ SLR protocol
- Conducting the review
- Reporting the review

3.1.2 Conduction of SLR

We followed the SLR standards when doing our SLR. First, we create SLR procedure to carry out our study. A senior professional from the CS/IT department reviewed it. The next sections go over the various steps of the SLR protocol.

A. SEARCH METHODS

A quest was conducted using phrases relating to the topic at hand. To generate a search string that addressed the research issue, keywords were composed. After that, the search string was used in a variety of search engines and academic libraries.

3.1.3 Search String

In order to design our SLR protocol, we followed the SLR standards. In addition, we used to locate keywords and build search strings, use the PICO criteria (Population, Intervention, Comparison, and Outcomes), based on the study objectives. The same techniques have been utilized by other researchers [8].

Population: Large-Scale Agile

Intervention: challenges, practices

Comparison: For the current investigation, no comparisons are made.

Relevant end-results: Knowledge sharing challenges and its practices in LSASD.

We have constructing the search string. ("Agile methods" OR "agile software development") AND ("Large-Scale Agile Development") AND ("knowledge sharing" OR "knowledge networks") AND ("challenges" OR "risks" OR "issues").

3.1.4 literature resources

Using our custom search term, we looked for libraries such as Google Scholar, Springer Link, IEEE Xplore, ACM, and Science Direct. Table 3.1 shows the search string results from several libraries.

B. CRITERIA FOR STUDY SELECTION

The scholars supported the authors' proposals as well as the strings. Initially, the string was appended to library metadata. The same was done with care to ensure that the title, abstract, and keyword requirements were not disturbed. Every paper was located and documented in great detail, along with the first writer keeping a thorough record. Other authors examined the articles based on this phase, assigning pertinent information pieces to each item, such as the abstract and title. Below the SLR conduction standards above, we established the following inclusion and exclusion criteria [8].

1) INCLUSION CRITERIA

- IC1: Full-text article in the English language.
- IC2: Sources that are relevant to LSASD.
- IC3: Sources that focus on KS challenges.
- IC4: Sources that focus on KS practices.
- IC5: Journal and conference papers, standards and white papers, and reports published by reputable organizations.

2) EXCLUSION CRITERIA

- EC1: Studies that are not related to our research questions.
- EC2: Papers other than the English language.
- EC3: Papers with workshop summaries.
- EC4: Duplicate sources
- EC5: Books, web pages, and magazine articles

Table 3. 1: Search string outcomes per database

Search String	Digital Libraries	Total Results	Initials Selection	Final Selection
("Agile methods, OR "agile software development") AND ("Large-Scale Agile Development") AND ("knowledge sharing" OR knowledge networks") AND ("challenges" OR "risks" OR "issues")	Google Scholar	213	45	35
	Science Direct	08	07	03
	ACM	12	03	01
	Springer Link	06	06	06
	IEEE Explore	09	05	03
Total		**248**	**66**	**48**

Table 3. 2: Assessment of quality for publication

ID	Quality Assessment Criteria	Answer
Q1	Does the paper provide well-defined aims and objectives?	Yes/no/Partially
Q2	Does the article have a clear context. e.g. industry or laboratory setting?	Yes/no/Partially
Q3	Does the paper explicitly discuss the limitations?	Yes/no/Partially
Q4	Does the paper add challenges in LSASD?	Yes/no/Partially
Q5	Does the paper add practices for KS in LSASD?	Yes/no/Partially

The outcomes of the search process are summarized in Table 3.1. We extracted 248 papers based on the inclusion criteria. We finally choose 48 papers after applying the exclusion criteria. The primary author accomplished each phase of the data extraction technique, which was then thoroughly evaluated by secondary authors.

C. QUALITY ASSESSMENT

Based on the publishing findings as well as the responses to our study questions, we graded the primary papers we looked at. Our criteria for determining quality are based on a few carefully chosen studies. Table 3.2 contains the questions for the quality criterion. We used a three-tier scale for quality assessment criteria to rate each question in the papers. Yes, No or partially is the answer. We assigned the values 2 Yes, 1 partly, and 0 No to produce measurable results.

In addition, the work was only included on the condition that it obtained a 0.5 average score. The first author was in responsibility of implementing the paper quality assessment criteria, while the other authors evaluated the comparable assessments on a slighter class of randomly selected analyzed. Six manuscripts were turned down by the second reviewer based on these quality standards. Deliberations were used to resolve any discrepancies. The quality assessment's major purpose was to weed out low-quality studies and determine the consistency of a study's findings [9].

D. EXTRACTION OF DATA

Each of the 48 articles yielded the following information:

- IDs of Paper and Titles
- Channel of Journal
- Publication Time
- Research Methodology

E. Synthesis of Data

The primary assessor removes information after the finalized journals. The data is double-checked by a secondary reviewer who chooses extraction forms at random. Changes are highlighted, and the principal reviewer implements the recommendations. Finally, the data extraction forms revealed 09 challenges. In addition, 39 practices for the 09 challenges are obtained.

3.1.5 Empirical Study

To verify the results of the SLR, we conducted a survey. We use the survey approach to make the most of the participation and accessible resources belong to various a questionnaire for countries will be devised for the collecting of knowledge. We'll conduct an online questionnaire survey to gather data from a variety of sources experts in

LSASD. The reason for that is to perform an online questionnaire survey collect data by enlisting the help of relevant specialists to supplement the research by investigating and validating the finding challenges and their practices in LSASD. Other scholars have utilized a related research method, SLR followed by an industrial survey. We used Google Forms to create and distribute our survey, which is a free platform for survey creation and distribution. Our goal was to confirm the SLR findings and identify any additional issues or practices that weren't currently known. In the following sections, we'll go over how to conduct a survey in a nutshell [9].

3.1.6 Questionnaire Design

At Malakand University, we established a questionnaire survey with the purpose of confirming SLR results from practitioners. We also plan to look for any additional practices not already recognized. The results of SLR are utilized as a source of information for the survey questionnaire. There are four sections to it. In section 1, a quick review of the study was comprised to provide experts with a general overview of the subject. The 2nd component is for gathering demographic information on experts. The difficulties found through SLR are presented in Section 3. To acquire quantitative data, the practitioners were given challenges to complete. The difficulties were assessed using a 5-point Likert scale. SLR practices were used as an input in section 4 and were rated on a Likert scale of 1 to 5. We used both open-ended and closed-ended questions to assess explicit knowledge. Participants were asked to rate their satisfaction with every issue and practices on a Likert scale of 1 to 5 (Strongly Agree, Agree, Not Sure, Disagree, Strongly Disagree). We questioned the experts whether there were any further problems in addition to the ones listed in order to obtain their feedback. Experts from SERG-UOM 4 at the University of Malakand evaluated the questionnaire. Their questions were answered. The QS were sent out after the pilot. Appendix C contains the whole questionnaire survey.

(i) Designing an Online Survey Questionnaire

Our research is divided into two sections: selection, locating relevant individuals, and developing the questions. The process of selecting persons to complete the required questionnaire is known as sampling. Participants (samples) must answer a series of questions in a questionnaire strategy. The next sections delve deeper into both of these subjects.

Sampling

A systematic approach and a non-systematic method to sampling are two approaches. For a small-scale survey, a non-systematic approach is employed because the list of an entire people is accessible Based on the statistic, samples are taken from that list. Because gathering contacts from a large number of businesses, producing a list of all members, and picking the best candidates is time-consuming, individuals from that incline is challenging, we adopted a nonsystematic technique to conduct our survey. A similar method has been used by other researchers. We adopted a strategy in the non-systematic sample sampling procedure [9]. The questionnaire distribution approach began with the creation of an invitation letter that included a summary of the study and was sent to the following websites.

1. LinkedIn Groups (www.linkedin.com).
2. We also invited for participation the authors of the industry papers selected through the SLR. The emails were published in the papers.. Details are given in Table 4

Table 3. 3: Summary of Agile Software Development Professional groups

S.NO.	Group Name	Channel	Members
1	Agile Scrum, Safe Transformation	LinkedIn	20,504
2	Agile, UX	LinkedIn	29,184
3	Scaled Agile Framework	LinkedIn	64,899
4	Agile CMMI	LinkedIn	18,872
5	Group of IT Professionals	LinkedIn	87,180
6	IT/Software Development Outsourcing and Offshoring	LinkedIn	28,947
7	Software Development outsourcing to Pakistan	LinkedIn	289
8	Outsourcing and Offshoring	LinkedIn	33,624
9	Agile	LinkedIn	76,819
10	Agile Lean Europe (ALE)	LinkedIn	6,061
11	Agile World Group	LinkedIn	29,318
12	Agile-Spain	LinkedIn	6,711
13	Agile and Lean Software Development	LinkedIn	191,189

3.1.7 Executing Online Questionnaire Survey

We invited all the 54 participants and asked them to answer the questionnaire study. In which, we acknowledged a total of 54 replies.

3.1.8 Data Analysis

To ensure the quality of the survey results, all entering questionnaires were assisted in deleting low-quality responses based on our pre-defined criteria:
Condition 1: Respondents having a small level of education, such as those with only an undergraduate degree.
Condition 2: A survey form that is unfinished or incomplete.
Condition 3: Multiple replies
An over-all of 54 replies were submitted by participants from all corners of the globe. As we applied our value criterion, two responses have been deleted. The information from the remaining 52 replies was assessed.

3. Results and Discussions

4.1 Results of SLRs

We used SLR to answer the first research question in this study. We focus on finding barriers/challenges faced by Large-Scale Agile Software Development organization. We also focus on providing the appropriate solutions/practices to mitigate those barriers/challenges.

4.1.1 Knowledge Sharing Challenges faced in Large-Scale Agile Software Development Organization.

We thoroughly review all the finally selected papers and dug out 09 critical challenges faced by Large-Scale Agile Software Development organizaiton, as shown in Table 4.1.

Table 4. 1: Knowledge sharing challenges in Large-Scale Agile Software Development

S. No	Challenges	Paper ID	Frequency	Percentage
1	Lack Of Process maturity	**P1, P3, P4, P5, P6, P7, P8, P12, P16, P24, P29, P34, P39, P40, P42, P44, P45, P46, P47, P48,**	20	41%
2	High degree of tasks dependencies	**P1, P9 P10, P15, P17, P23, P24, P26,P32,P33, P39,P40,P42,P43, P45,P46,P47,**	17	35%
3	Lack of effective management support	**P5, P11, P22, P26, P33, P35, P37, P39, P40, P42, P44, P45, P46, P47**	14	29%
4	Lack of strong collaboration among team members	**P2, P5, P11, P17, P19, P22, P23, P24, P28, P29, P37, P40, P45, P46**	14	29%
5	In-sufficient team efficiency	**P1, P3, P4, P5, P6, P7, P8, P12, P16, P24, P29, P34, P37, P39, P40, P42, P44, P45, P46, P47, P48,**	20	43%
6	Misinterpretation of Knowledge	**P1, P3, P4, P5, P6, P7, P8, P12, P16, P22, P24, P29, P34, P37, P39, P40, P42, P44, P45, P46, P47, P48,**	22	45%
7	Improper Knowledge Management	**P1, P3, P4, P5, P6, P7, P8, P9, P12, P16, P22, P24, P28, P29, P34, P37, P39, P40, P42, P44, P45, P46, P47, P48,**	24	50%
8	Multi Technological Environment	**P03, P06, P08, P12, P26, P27, P34, P38, P41, P43, P47**	11	23%
9	Team Diversity	**P02, P06, P16, P23. P31, P42, P43, P46**	08	16%

KSC#1 Lack of Process Maturity:

The increased elasticity that ASD promises comes with the cost of highlighting confidential communication between team members over written documentation or loss of the "big picture" of the product due to extensive focus on developing features. As a result, managing the knowledge asset becomes critical, and the lack of it can lead to several negative effects. For example, barriers to collaboration and asynchronous communication in large companies that introduced ASD and competitiveness loss [14].

KSC#2 High Degree of Task Dependencies:

The organizational structures that enable individuals to achieve a common goal. Interdependencies include resource sharing, activity synchronization, and prerequisite activities [5].

Working in unpredictable environments has a significant impact on how we organize our work. Managing uncertainty necessitates constant adaptation and change management. As a result, organizations must strike a balance between adaptability and stability [15].

KSC#3 Lack of effective management support:

In Large-Scale Agile Software Development Organization coordination between team and team members is very important factor [16].

Self-organizing and self-sufficient agile software development teams have the power to make decisions about how they work internally [17].

The organisation had to deal with a number of cross-team difficulties, including how to coordinate growth between teams, how to enhance specific competencies across teams, and how to enable the necessary cultural shift [17].

KSC#4 Lack of strong collaboration among team members:

The organisation had to deal with a number of cross-team difficulties, including how to coordinate growth between teams, how to enhance specific competencies across teams, and how to enable the necessary cultural shift [18].
Agile approaches are increasingly being used in large-scale software development. A team's alignment with other teams and the rest of the organization is critical in large-scale initiatives. This has been demonstrated to jeopardize team autonomy, lowering responsiveness and flexibility [18].

KSC#5 Insufficient Team Efficiency:
Large-scale agile software development is complicated and fraught with difficulties. Large projects necessitate effective team coordination and communication, the management of interteam interdependence, the involvement of nonagile groups, and the inclusion of the proper personnel [9].
To identify the importance of scaling strategies, difficulties, and success factors, we perform a concentrated literature review. This concentrated literature review's findings are being utilised to guide action research within a software organization with the goal of scaling agile processes [9].
KSC#6 Misinterpretation of Knowledge:
Due to many teams involved in Large-Scale Agile Software Development Organization, misunderstanding of knowledge occur between team and team members. Mitigation measures used by practitioners include informal conversation, cultural exchange, a shared platform, tools, visual prototyping, shared chat rooms, rotation, and overlapping hours. Proper recruitment process, orientation and training session, Coaching and Support to overcome this challenge [19].

KSC#7 Improper Knowledge Management:
Improper Knowledge Management is also a big challenge for Large-Scale Development Organizations. To over come these challenges two type of factors needed, Task uncertainty refers to the difficulty and variability of the work that an organisational unit performs. Increased task uncertainty is indicated by higher levels of complexity, thinking time to solve problems, or time necessary before an outcome is known. Task interdependence is the degree to which members of an organisational unit rely on others to do their tasks. When there is a lot of task-related collaboration, there is a lot of interdependence [5].
KSC#8 Multi Technological Environment:
A suitable infrastructure for communication, information exchange, and a community of practise can assist the development of knowledge network and social capital in a large-scale project. To enable end-to-end development, the infrastructure is required. This includes collaboration tools, test environments, continuous integration, and automated deployment [20].
KSC#9 Team Diversity:
A lack of the correct attitude and culture, according to several research, prevents businesses from realising the full potential of SAFe, LeSS, other scaled methodologies. To achieve the organisational culture, the transformation must begin with people's mindsets. Transparency and ongoing improvement based on experimentation necessitate courage and a mentality shift [2020].

4.1.2 Analysis of the knowledge sharing challenges faced by large-scale agile software development

This section includes the results of our SLR, we conducted statistical analysis on the highlighted challenges. These criteria include the paper's continent/country and study strategy. The goal of these studies is to determine if these obstacles are consistent/uniform between continents and study strategies, or vice versa.

A. Analysis of the knowledge sharing challenges faced by large-scale agile software development based on continent

The challenges found in each continent are listed in Table 4.2. Figure 4.1 shows the number of articles/reporting studies published in each continent. Only the challenges listed in two continents (Asia and Europe) were

compared to those listed in other continents (combination of two or more continents). Our goal is to determine whether these issues differ by continent or are universal. The data type ordinal is present in the SPSS data set. Significant differences between the obstacles mentioned on different continents were discovered using the linear by linear association Chi-square test.

Table 4. 2: Summary of the knowledge sharing challenges faced by large-scale agile software development

Challenges	**Occurrence in SLR (N=48)**						**Chi-square Test (Linear-by-Linear Association) a = .05**	
	Asia (N=10)		Europe (N=34)		Others (N=4)		X^2	P
	Freq	**%**	**Freq**	**%**	**Freq**	**%**		
Lack Of Process maturity	5	10%	14	41%	1	2%	.684	.408
High degree of tasks dependencies	3	6%	13	27%	1	2%	.005	.943
Lack of effective management support	5	10%	9	18%	0	0%	3.778	.052
Lack of strong collaboration among team members	3	6%	11	22%	0	0%	.559	.455
In-sufficient team efficiency	6	12%	14	29%	1	2%	1.694	.193
Misinterpretation of Knowledge	6	12%	15	31%	1	2%	1.507	.220
Improper Knowledge Management	6	12%	17	35%	1	2%	1.182	.277
Multi Technological Environment	4	8%	6	12%	1	2%	1.105	.293
Team Diversity	3	6%	3	6%	2	4%	.000	1.000

When comparing ordinal variables, the Pearson Chi-square test is preferred above the linear association test by linear association test since it is more powerful. The findings in Table 4.2 show that there are further parallels than variances among the issues between continents. Table 4.2 demonstrates that Europe has all of the issues, while Asia has less challenges. “Lack of effective management support” and “Strong collaboration among the agile team” others contain 0 occurrences.

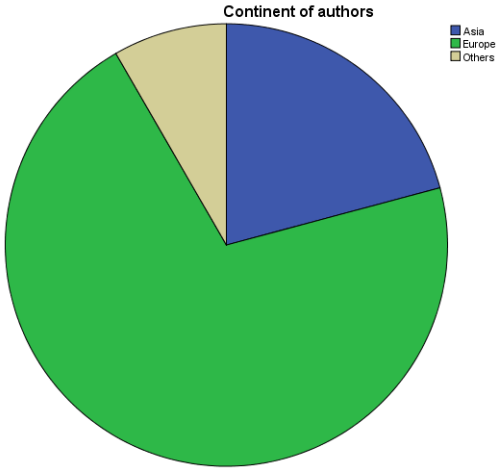

Figure 4. 1: Analysis based on continent

In figure 4.1, shows analysis based on continent. Green color indicates the continent Europe, blue color indicate Asia and the gray color indicate other continents. This figure demonstrates that Europe has all of the issues while in other continents have less challenges than Europe.

A. Analysis of the Knowledge sharing Challenges faced by LSASD, Identified through SLR, Based on decades

Table 4.3: Analysis of the Knowledge sharing challenges faced by LSASD based on decades

Challenges	**OCCURRENCE IN SLR(N=48)**				**Chi-square Test (Linear-by-Linear Association) a = .05**	
	From 2010-2015 (N=8)		From 2016-2022 (N=40)		X^2	P
	Freq	**%**	**Freq**	**%**		
Lack of Process maturity	5	10%	15	30%	.074	.786
High degree of tasks dependencies	6	12%	11	22%	.880	.348
Lack of effective management support	3	6%	11	22%	.313	.576
Lack of strong collaboration among team members	5	10%	9	18%	.730	.393
In-sufficient team efficiency	5	10%	16	32%	.198	.656
Misinterpretation of Knowledge	6	12%	16	32%	.001	.979
Improper Knowledge Management	6	12%	18	36%	.103	.748
Multi Technological Environment	3	6%	8	16%	.000	.987
Team Diversity	3	6%	5	10%	.516	.472

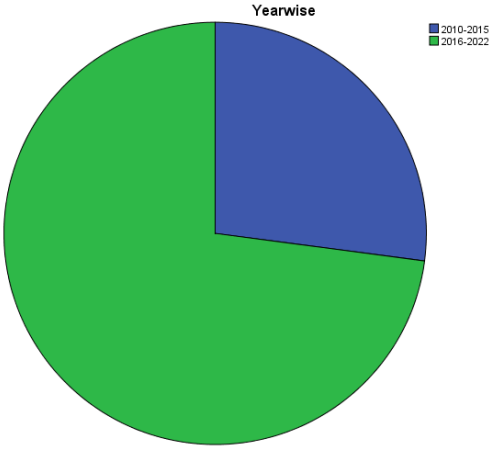

Figure 4.2: Analysis based on decades

Figure 4.3 describes a comparison of two periods' problems. Green color shows period from 2016-2022 and blue color shows the period of 2010-2015. During the implementation of our systematic literature review search, we did not set any date limits. However, as previously stated, LSASD contributions began in 2009. The papers have been grouped into five years. The first period is (2010-2015), while the second term is (2016-2022). Because the majority of publications are published in period two, such as those displayed in Figure 4.3, we may conclude that LSASD is still one of the most critical fields of research.

4.2 Practices/Solutions of Challenges in LSASD

To address the nine difficulties outlined, 39 practises were identified. The rest of the tables, from Table 4.6 to Table 4.14, contain more information about these activities.

Table 4.4: Practices for addressing lack of process maturity

C #1 Lack of Process Maturity		
S.No	Practices to address the Lack of Process Maturity	Frequency
KSCP#1.1	Need a proper decision making process	7
KSCP#1.2	Focus on the efforts to manage organization's workforce	7
KSCP#1.3	Proper creation, storage, retrieval, transfer of knowledge is required	3
KSCP#1.4	Agile team should apply various mitigation techniques, common platform, tools, visual prototyping, common chat rooms and overlapping hours.	3
KSCP#1.5	Focus on quality of a software	9
KSCP#1.6	To establish a holistic map of challenges and candidate solutions make re-engineering more efficient within agile environments.	7
KSCP#1.7	Focus continuously on process improvement as it might differ for the different projects	8

Table 4. 5: Practices for addressing High Degree of Task Dependencies

C#2 High Degree of Task Dependencies		
S.NO.	Practices for addressing High Degree of Task Dependencies	Frequency
KSCP#2.1	Examine agile approaches that operate as coordination mechanisms in a large-scale project using a dependency taxonomy.	9
KSCP#2.2	Need a balance among the agile team and the organization agility	6
KSCP#2.3	Tasks uncertainties should be evaluated and must be characterized, to be assigned to appropriate agile developer	2
KSCP#2.4	Improve collaboration between agile teams, to improve dependency management and enhance knowledge transparency	6
KSCP#2.5	Plan Meetings, impromptu discussions, communication tools, and an open work area with bulletin boards provided a vital platform for managing dependencies.	3

Table 4.6: Practices for addressing Lack of efficient management support

C#3 Lack of Efficient Management support		
S.NO.	Practices for Addressing Lack of Efficient Management Support	Frequency
KSCP#3.1	Rotation of agile team members on requirement basis	4
KSCP#3.2	Bringing developers early in the picture for project design decisions	5
KSCP#3.3	Update the agile team through Internal & External trainings	4
KSCP#3.4	Strong and dedicated management and leadership support is required	8
KSCP#3.5	New strategies need to be developed for enhancing competencies and capabilities	5

Table 4. 7: Practices for addressing the Lack of strong collaboration among the agile team

C#4 Lack of strong collaboration among the agile team		
S.NO.	Practices for Addressing Lack of strong collaboration among the agile team	Frequency
KSCP#4.1	Coordinate with the agile team on regular and detailed basis	5
KSCP#4.2	Adapt Development and operation functions through a tight integration	3
KSCP#4.3	Coordination mechanisms should be modified to accommodate large-scale agile software development	5
KSCP#4.4	Plan Meetings, impromptu discussions, communication tools, and an open work area with bulletin boards provided a vital platform for managing dependencies.	3

Table 4. 8: Practices for addressing insufficient team efficiency

C#5 Insufficient team efficiency		
S.NO.	Practices for Addressing insufficient team efficiency	Frequency
KSCP#5.1	Talented agile team must be hired and should be trained internally and externally on regular basis	4
KSCP#5.2	Rotation of agile team members on requirement basis	7
KSCP#5.3	New strategies need to be developed for enhancing competencies and capabilities	11
KSCP#5.4	Bringing developers early in the picture for project design decisions	6

Table 4. 9: Practices for addressing Misinterpretation of Knowledge

C#6 Misinterpretation of Knowledge		
S.NO.	Practices for Addressing Misinterpretation of Knowledge	Frequency
KSCP#6.1	Measure cognitive and psychological distance between the agile teams using interactive posters	7
KSCP#6.2	Create common chat rooms and use common platforms for knowledge sharing	3
KSCP#6.3	Plan Meetings, impromptu discussions, communication tools, and an open work area with bulletin boards provided a vital platform for managing dependencies.	3

Table 4. 10: Practices for addressing Improper Knowledge Management

C#7 Improper Knowledge Management		
S.NO.	Practices for addressing Improper Knowledge Management	Frequency
KSCP#7.1	Proper creation, storage, retrieval, transfer of knowledge	3
KSCP#7.2	Needs a well-planned design structure for knowledge sharing and management	9
KSCP#7.3	Prioritize the user stories accurately	9
KSCP#7.4	Focus on quality attributes	9

Table 4. 11: Practices to address Multi Technological Environment

C#8 Multi Technological Environment		
S.NO.	Practices to Addressing Multi Technological Environment	Frequency
KSCP#8.1	Uniform modeling language should be used to illustrate the software architecture and high level requirements	4
KSCP#8.2	Uniform Technology and well-structured information needs to be used for knowledge sharing systems	8
KSCP#8.3	Create common chat rooms and use common platforms for knowledge sharing	3

Table 4. 12: Practices for addressing Team Diversity

C#9 Team Diversity		
S.NO.	Practices for Addressing Team Diversity	Frequency
KSCP#9.1	Agile team must use informal communication, cultural exchange, common platform, tools, visual prototyping, common chat rooms, rotation, and overlapping hours	9
KSCP#9.2	Measure cognitive and psychological distance between the agile teams using interactive posters	1
KSCP#9.3	Translate between the business language used by the customer and the technical language employed by the team	7

4.2.1 Conduction of Empirical Study

We surveyed experts to evaluate the challenges in adopting and to acquire just about practices to those challenges. A survey is a form of empirical investigation in which questions are asked and data is gathered to create a quantitative or numeric narrative of a percentage of the population or sample. Other researchers have taken a similar strategy.

4.2.2 Data Analysis

To ensure the quality of the survey results, all entering questionnaires were assisted in deleting low-quality responses based on our pre-defined criteria:

Standard 1: Respondents having a low small of education, such as those with only an undergraduate degree.

Standard 2: An unfinished or partially completed questionnaire.

Standard 3: Multiple responses

Contestants from all there were 54 answers from all over the globe. wo responses were removed when we used our quality criterion. The remaining data sources yielded 52 replies, which were evaluated.

4.3 Challenges faced by LSASD identified through empirical study

Table 4. 13: Challenges faced by LSASD identified through empirical study

Challenges	Experts Response (n = 52)							
	Positive			Negative			Not Sure	
	Agree	Strongly Agree	%	Disagree	Strongly Disagree	%	Not sure	%
Lack of Process Maturity	42	7	94%	3	0	5%	0	0%
High Degree of Tasks dependencies	41	5	88%	5	1	11%	1	1%
Lack of Efficient Management Support	33	12	86%	3	2	9%	2	3%
Lack of Strong Collaboration among Agile Team	28	23	98%	1	0	1%	0	0%
Insufficient Team Efficiency	37	6	82%	4	4	15%	1	1%
Misinterpretation of Knowledge	34	4	73%	3	5	15%	6	11%
Improper Knowledge Management	37	9	88%	4	1	9%	1	1%
Multi Technological Environment	29	16	86%	4	1	9%	2	3%
Team Diversity	29	5	65%	7	1	15%	10	19%

To respond to RQ2, Table 4.14 summarizes the issues that our empirical research revealed. According to the findings, seven of the nine there is an occurrence of problems. of >80%, whereas the remaining two have a high rate of >60% on the good side of things. The utmost common occurrence is “lack of strong collaboration among agile team”. It occurs 98% of the time. Users are eager for a service that meets their expectations in terms of quality. LSASD providers must ensure that services are delivered quickly and with the needed level of quality. Our findings also show that the “lack of process maturity” is the second most common event, accounting for 94 percent of the time. Users also seek assurances regarding the services that are dynamically offered to them, both during high and low demand periods. Vendors must ensure that services are not affected as a result of load fluctuations.

According to our findings, the third most desirable occurrence is “High degree of task dependencies” and “Improper Knowledge Management” which occurs 88 percent of the time. The needs of users should be met per their criteria. Vendors must ensure that services are delivered per the contract's user expectations. This will foster a relationship of trust between vendors and clients. Other frequently mentioned positive challenges include: “Lack of efficient management support” (86%), “Multi Technological Environment” (86%), and “Insufficient team efficiency” (82%), “Misinterpretation of Knowledge” (73%) and “Team Diversity” (65%).

All challenges have been listed in Table 4.14 neutral list (not sure). This demonstrates that these professionals are unaware of the difficulties. The #1 most commonly listed challenge on the neutral list is "Team Diversity" that gain 19% of the over-all. The 2nd most mentioned problem is "Misinterpretation of Knowledge," that gain 11% of the over-all, and the third most mentioned challenge is "Lack of

efficient management support and "Multi Technological" that gain 3% of the over-all.

4.4 Analysis of the Challenges faced by LSASD identified through empirical study

We achieved different data investigations gather after 52 supporters. Our aim was to gain concrete understanding of the responses collected from various people. We analyzed it after each side wi'll do everything we can to cover every perspective. We performed 4 analysis on our realistic outcomes, i.e analysis based on company size, continent, experience of the participants and scope of the company. They are discussed in details in the following sub sections.

Table 4. 14: Analysis based on experience of the participants

S. No	Challenges	EXPERT RESPONSE (N=52)									Chi-Square Test (Linear-by-Linear associationα = 0.05), df=1	
		1-5 year (N=19)			6-10 year (N=21)			>10 year (N=12)				
		A	DA	NS	A	DA	NS	A	DA	NS	X^2	P
1	Lack Of Process maturity	18	1	0	21	0	0	10	2	0	1.518	.218
2	High degree of tasks dependencies	17	2	0	20	1	0	8	3	1	2.119	.145
3	Lack of effective management support	16	2	1	19	1	1	9	3	0	.387	.534
4	Lack of strong collaboration among team members	18	0	1	21	0	0	12	0	0	.303	.582
5	In-sufficient team efficiency	17	2	0	20	1	0	6	5	1	5.008	.025
6	Misinterpretation of Knowledge	13	3	3	18	2	1	6	4	1	.425	.514
7	Improper Knowledge Management	18	0	1	20	0	1	12	0	0	.111	.739
8	Multi Technological Environment	15	3	1	20	0	1	11	1	0	3.127	.077
9	Team Diversity	16	3	0	13	2	6	5	3	4	3.393	.065

In Figure 4.5 we have mentioned empirical analysis based on employee experience. Green color shows that an employee have 6 to 10 years' experience. The blue color indicates that an employee having experience from 1 to 5 years and the gray color shows that an employee having experience of more than 10 years.

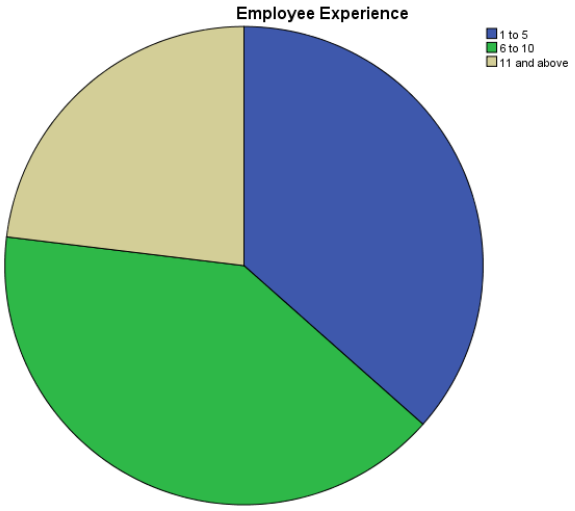

Figure 4.3: Empirical Analysis based on Employee Experience

Table 4. 15: Analysis based on continent

S. No	Challenges	EXPERT RESPONSE (N=52)									Chi-Square Test (Linear-by-Linear associationα = 0.05), df=1	
		Asia (N=33)			Europe (N=11)			Others (N=8)				
		A	DA	NS	A	DA	NS	A	DA	NS	X^2	P
1	Lack Of Process maturity	30	3	0	11	0	0	8	0	0	2.878	.090
2	High degree of tasks dependencies	27	5	1	10	1	0	8	0	0	2.000	.157
3	Lack of effective management support	25	6	2	11	0	0	8	0	0	4.470	.035
4	Lack of strong collaboration among team members	32	0	1	11	0	1	8	0	0	1.044	.307
5	In-sufficient team efficiency	25	7	1	10	0	1	8	0	0	4.465	.035
6	Misinterpretation of Knowledge	21	8	3	8	1	2	8	0	0	3.001	.083
7	Improper Knowledge Management	32	0	1	10	0	1	8	0	0	.075	.785
8	Multi Technological Environment	27	4	2	11	0	0	8	0	0	1.167	.280
9	Team Diversity	**17**	**8**	**0**	**9**	**0**	**2**	**8**	**0**	**0**	**9.652**	**.002**

In Figure 4.6 we have mentioned empirical analysis based on country/continent wise. Blue color shows continent Asia and having more responses have received. Green color indicates Europe and less responses received than Asia. The gray color indicates other continents and less responses received from other continents.

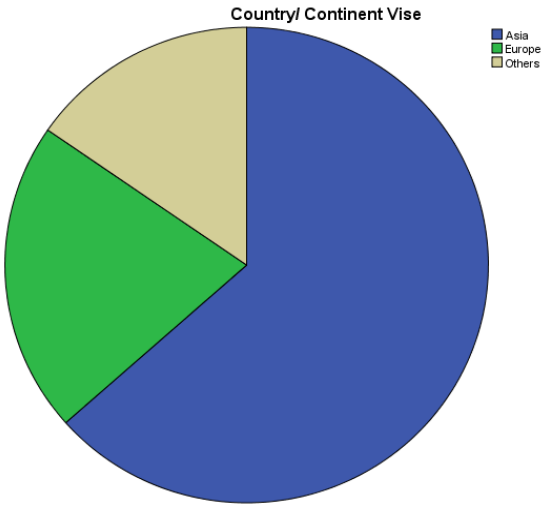

Figure 4.4: Empirical Analysis based on continent vise

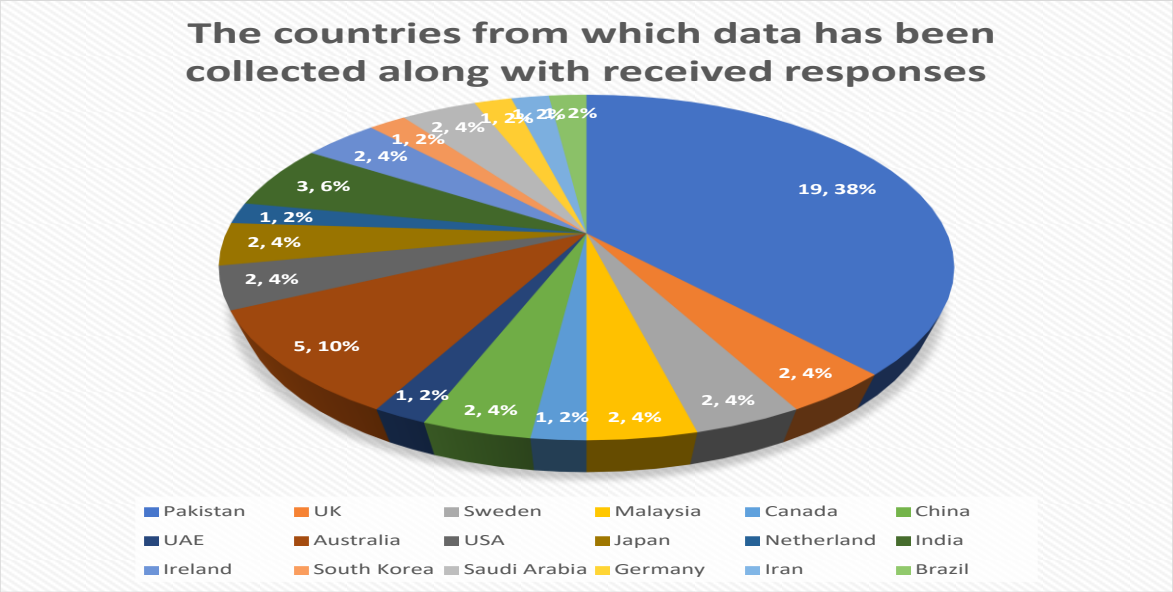

Figure 4. 5: Countries from which data has been collected

Empirical Practices for the adoption of LSASD

Through empirical research, 39 practices were discovered to address the nine stated problems. As input to our questionnaire survey, we use practices found through SLR. Tables 4.18 to 4.26 show the practices discovered through empirical research.

Table 4. 16: Practices for addressing Lack of Process maturity

KSC #1 Lack of Process Maturity		**Positive**			**Negative**			**Not Sure**	
S.NO	Practices to address the Lack of process maturity	A	SA	%	D	SD	%	NS	%
KSCP#1.1	Need a proper decision making process	40	8	92%	4	0	7%	0	0%
KSCP#1.2	Focus on the efforts to manage organization's workforce	39	6	86%	2	2	7%	3	5%
KSCP#1.3	Proper creation, storage, retrieval, transfer of knowledge is required	39	6	86%	3	2	9%	2	3%
KSCP#1.4	Informal communication, cultural interchange, a shared platform, tools, visual prototyping, common chat rooms, rotation, and overlapping hours are all required of agile teams.	28	20	92%	4	0	7%	0	0%
KSCP#1.5	Focus on quality of a software	28	20	92%	2	2	7%	0	0%
KSCP#1.6	To establish a holistic map of challenges and candidate solutions make re-engineering more efficient within agile environments.	43	1	84%	4	1	9%	3	5%
KSCP#1.7	Focus continuously on process improvement as it might differ for the different projects	34	11	86%	5	1	11%	1	1%

Table 4. 17: Practices for addressing High degree of task dependencies

KSC #2 High Degree of Task Dependencies		Positive			Negative			Not Sure	
S.NO	Practices for addressing High Degree of Task Dependencies	A	SA	%	D	SD	%	NS	%
KSCP#2.1	Examine agile approaches that operate as coordination mechanisms in a large-scale project using a dependency taxonomy.	38	4	80%	7	0	13%	3	5%
KSCP#2.2	Need a balance among the agile team and the organization agility	37	7	82%	7	0	13%	1	1%
KSCP#2.3	Tasks uncertainties should be evaluated and must be characterized, to be assigned to appropriate agile developer	35	3	73%	5	3	15%	6	11%
KSCP#2.4	Improve collaboration between agile teams, to improve dependency management and enhance knowledge transparency	24	20	84%	6	1	13%	1	1%
KSCP#2.5	Plan Meetings, impromptu discussions, communication tools, and an open work area with bulletin boards provided a vital platform for managing dependencies.	29	4	63%	4	4	15%	11	21%

Table 4. 18: Practices for addressing Lack of Efficient Management Support

KSC #3 Lack of Efficient Management Support		Positive			Negative			Not Sure	
S.NO	Practices for addressing Lack of Efficient Management Support	A	SA	%	D	SD	%	NS	%
KSCP#3.1	Rotation of agile team members on requirement basis	36	3	75%	8	0	15%	5	9%
KSCP#3.2	Bringing developers early in the picture for project design decisions	37	3	76%	5	2	13%	5	9%
KSCP#3.3	Update the agile team through Internal & External trainings	25	18	82%	5	2	13%	2	3%
KSCP#3.4	Strong and dedicated management and leadership support is required	28	18	88%	2	4	11%	0	0%
KSCP#3.5	New strategies need to be developed for enhancing competencies and capabilities	21	23	84%	3	4	13%	1	1%

Table 4. 19: Practices for addressing Lack of Strong Collaboration among Agile Team

KSC#4 Lack of Strong Collaboration among Agile Team		Positive			Negative			Not Sure	
S.NO	Practices for addressing Lack of Strong Collaboration among Agile Team	A	SA	%	D	SD	%	NS	%
KSCP#4.1	Coordinate with the agile team on regular and detailed basis	27	23	96%	2	0	3%	0	0%
KSCP#4.2	Adapt Development and operation functions through a tight integration	45	3	92%	2	0	3%	2	3%
KSCP#4.3	Coordination mechanisms should be modified to accommodate large-scale agile software development	18	31	94%	2	0	3%	1	1%
KSCP#4.4	Plan Meetings, impromptu discussions, communication tools, and an open work area with bulletin boards provided a vital platform for managing dependencies.	32	3	67%	2	1	5%	14	26%

Table 4. 20: Practices for addressing In-sufficient Team Efficiency

KSC #5 In-sufficient Team Efficiency		Positive			Negative			Not Sure	
S.NO	Practices for addressing In-sufficient Team Efficiency	A	SA	%	D	SD	%	NS	%
KSCP#5.1	Talented agile team must be hired and should be trained internally and externally on regular basis	37	8	86%	6	1	13%	0	0
KSCP#5.2	Rotation of agile team members on requirement basis	37	2	75%	5	3	15%	5	9%
KSCP#5.3	New strategies need to be developed for enhancing competencies and capabilities	19	26	86%	6	0	11%	1	1%
KSCP#5.4	Bringing developers early in the picture for project design decisions	36	5	78%	4	3	13%	4	7%

Table 4. 21: Practices for addressing Mis-Interpretation of Knowledge

KSC#6 Mis-Interpretation of Knowledge		Positive			Negative			Not Sure	
S.NO	Practices for addressing Mis-Interpretation of Knowledge	A	SA	%	D	SD	%	NS	%
KSCP#6.1	Measure cognitive and psychological distance between the agile teams using interactive posters	24	2	50%	9	0	17%	17	32%
KSCP#6.2	Create common chat rooms and use common platforms for knowledge sharing	32	1	63%	8	1	17%	10	19%
KSCP#6.3	Plan Meetings, impromptu discussions, communication tools, and an open work area with bulletin boards provided a vital platform for managing dependencies.	27	10	71%	7	4	21%	4	7%
KSCP#6.4	Translate between the business language used by the customer and the technical language employed by the team	35	8	82%	8	1	17%	0	0%

Table 4. 22: Practices for addressing Improper Knowledge Management

KSC#7 Improper Knowledge Management		Positive			Negative			Not Sure	
S.NO	Practices for addressing Improper Knowledge Management	A	SA	%	D	SD	%	NS	%
KSCP#7.1	Proper creation, storage, retrieval, transfer of knowledge	37	2	75%	4	1	9%	8	15%
KSCP#7.2	Needs a well-planned design structure for knowledge sharing and management	43	0	82%	5	1	11%	3	5%
KSCP#7.3	Prioritize the user stories accurately	31	11	80%	3	4	13%	3	5%
KSCP#7.4	Focus on quality attributes	14	32	88%	5	1	11%	0	0%

Table 4. 23: Practices for addressing Multi Technological Environment

KSC#8 Multi Technological Environment		Positive			Negative			Not Sure	
S.NO	Practices for addressing Multi Technological Environment	A	SA	%	D	SD	%	NS	%
KSCP#8.1	Uniform modeling language should be used to illustrate the software architecture and high level requirements	38	2	76%	4	1	9%	7	13
KSCP#8.2	Uniform Technology and well-structured information needs to be used for knowledge sharing systems	35	10	86%	4	3	13%	1	1%
KSCP#8.3	Create common chat rooms and use common platforms for knowledge sharing	33	4	71%	4	2	11%	9	17

Table 4. 24: Practices for addressing Team Diversity

KSC#9 Team Diversity		Positive			Negative			Not Sure	
S.NO	Practices for addressing Team Diversity	A	SA	%	D	SD	%	NS	%
KSCP#9.1	Informal communication, cultural interchange, a shared platform, tools, visual prototyping, common chat rooms, rotation, and overlapping hours are all required of agile teams.	36	5	78%	10	0	19	1	1%
KSCP#9.2	Measure cognitive and psychological distance between the agile teams using interactive posters	23	3	50%	9	1	19%	16	30%
KSCP#9.3	Translate between the business language used by the customer and the technical language employed by the team	33	8	78%	10	0	19%	1	1%

4.5 A Comparison of the challenges of Knowledge Sharing Challenges across Two Data Sets (Systematic Literature Review vs Questionnaire Survey)

This section compares the outcomes of SLRs with questionnaire surveys, in addition to the difficulties highlighted by SLR and their validation with QS. Such a comparison was intended to give a bit glow upon the similarities along with differences between the two pieces of data ' success determinants. Table 4.27 shows an overview of all the difficulties found by SLR and the QS.

The systematic literature review records has not been subjected to any kind of classification. Though, the information from the questionnaire is divided into four categories: Agree, Strongly Agree, Disagree, Strongly Disagree, and Not Sure. Table 4.27 is created to compare these two data sets using only Strongly Agree value of success factors from the questionnaire survey.

Table 4.27 Comparison of the Knowledge Sharing Challenges across SLR and Questionnaire Survey, for Large-Scale Agile Software Development

Table 4. 25: Knowledge sharing Challenges in Large-Scale Agile Software Development

S. No	Challenges	Occurrence in SLR frequency & % (N=48)		Agree frequency & % in the Questionnaire Survey	
		Frequency	%	Frequency	%
1	Lack Of Process maturity	20	41%	7	14%
2	High degree of tasks dependencies	17	35%	5	10%
3	Lack of effective management support	14	29%	12	25%
4	Lack of strong collaboration among team members	14	29%	23	47%
5	In-sufficient team efficiency	20	43%	6	12%
6	Misinterpretation of Knowledge	22	45%	4	8%
7	Improper Knowledge Management	24	50%	9	18%
8	Multi Technological Environment	11	23%	16	33%
9	Team Diversity	08	16%	5	10%

Several open-ended questions were added in the questionnaire to identify any unobserved issues in addition to the recognized ones, in order to gather tacit knowledge on Knowledge Sharing Challenges. However, no new difficulties were detected in the survey, which explains why the number of challenges identified in the two data sets in Table 4.27 is identical. It is also obvious from the empirical data in Table 4.27 that no challenge in the survey had a zero frequency.

5.1 Limitations of the research

We used a systematic literature review as a research approach, including the use of appropriate strings and a suitable sample, to the best of our ability. We can assume, however, that we may have omitted some vital information. The study's keywords were used after a thorough discussion and suggestions by the two writers to ensure the study's validity and contain as much relevant content as feasible.

To reduce the danger of construct validity, such as the use of digital libraries, we focused our

search on the most relevant and well-known libraries in computing disciplines. However, we only looked at a small number of libraries, thus increasing the risk of data misuse. We have mentioned the publications in Appendix A for the internal validity of research, from whom the data is retrieved to lessen the danger for interested viewers. To decrease the risk, the collaborators used a methodical methodology and intervallic checks to confirm each part of the SLR. The justification for this adjustment was because similar research have previously utilized the same approach. Another restriction or danger was that we only chose items composed in English language, which raised up the likelihood of repeated publications on the same subject. Furthermore, only articles written in English were chosen.

The Questionnaire Survey looked at agile professionals' perspectives and experiences with Knowledge Sharing Challenges and their practices in agile methodologies in Large-Scale Agile Software Development Organizations. This section has also looked into the hazards to the empirical study's validity. We received 54 responses to the online poll. We should have included more international participants for better results, but due sto a lack of resources and time, this was not feasible at the time. We've also reached out to overseas professionals with all of our tools, including encouraging them to join a number of LinkedIn agile groups. On the other hand, their participation was entirely voluntary. Because of the small number of overseas respondents, the results should be regarded with caution.

4. Conclusion

Agile methods are extensively adapted by the software development organizations due to its promising and appealing benefits and values that it adds to the software products and overall process of the software development. Agile at Large-Scale refers to the process of developing and delivering enterprise-class systems and software with a significant number of groups. Considering the widespread use of agile methods, large-scale agile software development has also adapted these methods to leverage its benefits. Large-Scale Agile development, on the other hand are confronted with significant challenges. Knowledge sharing plays a vital role in software development organization, and the software industry is said to require more knowledge management than any process. Knowledge Sharing is a valuable in agile software development organization that is transformed in to products during the development process. A significant amount of study has been done in the literature to determine the obstacles. Still, the research area is in its infancy, and more research is needed to uncover the issues and practices that the large-scale agile development is facing.

References

[1] S. Abdullahi and L. Bagiwa, "A Review on the Process of Adoptability of Agile Methods in Software Development Practices," American Journal of Engineering Research, pp. 199-207, 2019.

[2] N. Rashid and S. U. Khan, "Green agility for global software development vendors: A systematic literature review protocol," Proceedings of the Pakistan Academy of Sciences, vol. 52, pp. 301-313, 2015.

[3] M. F. Abrar, M. S. Khan, S. Ali, U. Ali, M. F. Majeed, A. Ali, et al., "Motivators for large-scale agile adoption from management perspective: A systematic literature review," Ieee Access, vol. 7, pp. 22660-22674, 2019.

[4] N. Sekitoleko, F. Evbota, E. Knauss, A. Sandberg, M. Chaudron, and H. H. Olsson, "Technical dependency challenges in large-scale agile software development," in International conference on agile software development, 2014, pp. 46-61.

[5] T. Dingsøyr, N. B. Moe, and E. A. Seim, "Coordinating knowledge work in multiteam programs: findings from a large-scale agile development program," Project Management Journal, vol. 49, pp. 64-77, 2018.

[6] F. Dumitriu, G. Meșniță, and L.-D. Radu, "Challenges and Solutions of Applying Large-Scale Agile at Organizational Level," Informatica Economica, vol. 23, 2019.

[7] E. Bjarnason, B. Gislason Bern, and L. Svedberg, "Inter-team communication in large-scale co-located software engineering: a case study," Empirical Software Engineering, vol. 27, pp. 1-43, 2022.

[8] A. Ahmad, S. U. Khan, H. U. Khan, G. M. Khan, and M. Ilyas, "Challenges and Practices Identification via a Systematic Literature Review in the Adoption of Green Cloud Computing: Client's Side Approach," IEEE Access, 2021.

[9] M. Ilyas and S. U. Khan, "An empirical investigation of the software integration success factors in GSD environment," in 2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA), 2017, pp. 255-262.

[10] R. Hoda and L. K. Murugesan, "Multi-level agile project management challenges: A self-organizing team perspective," Journal of Systems and Software, vol. 117, pp. 245-257, 2016.

[11] M. Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen, "Large-scale agile transformation at Ericsson: a case study," Empirical Software Engineering, vol. 23, pp. 2550-2596, 2018.

[12] A. de Bem Machado, S. Secinaro, D. Calandra, and F. Lanzalonga, "Knowledge management and digital transformation for Industry 4.0: a structured literature review," Knowledge Management Research & Practice, pp. 1-19, 2022.

[13] N. B. Moe, B. H. Dahl, V. Stray, L. S. Karlsen, and S. Schjødt-Osmo, "Team autonomy in large-scale agile," in Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS), 2019, pp. 6997-7006.

[14] R. A. B. Ouriques, K. Wnuk, T. Gorschek, and R. B. Svensson, "Knowledge management strategies and processes in agile software development: a systematic literature review," International journal of software engineering and knowledge engineering, vol. 29, pp. 345-380, 2019.

[15] F. Almeida, E. Miranda, and J. Falcão, "Challenges and facilitators practices for knowledge management in large-scale scrum teams," Journal of Information Technology Case and Application Research, vol. 21, pp. 90-102, 2019.

[16] N. B. Moe and T. Dingsøyr, "Emerging research themes and updated research agenda for large-scale agile development: a summary of the 5th international

workshop at XP2017," in Proceedings of the XP2017 Scientific Workshops, 2017, pp. 1-4.

[17] E. Tomé, E. Gromova, and A. Hatch, "Knowledge management and COVID-19: Technology, people and processes," Knowledge and Process Management, 2022.

[18] N. Rashid and S. U. Khan, "Using agile methods for the development of green and sustainable software: Success factors for GSD vendors," Journal of Software: Evolution and Process, vol. 30, p. e1927, 2018.

[19] F. O. Bjørnson and K. Vestues, "Knowledge sharing and process improvement in large-scale agile development," in Proceedings of the Scientific Workshop Proceedings of XP2016, 2016, pp. 1-5.

[20] J. Stevenson and M. Wood, "Recognising object-oriented software design quality: a practitioner-based questionnaire survey," Software Quality Journal, vol. 26, pp. 321-365, 2018.