Data Management in Agile Software Development: Challenges and Solutions

A thesis presented in partial fulfilment of the requirements for the degree of

Master of Information Sciences

in

software engineering

Massey University
Palmerston North, New Zealand.

Ahmed Mohamed

2024

Abstract

Managing data in agile software development poses significant challenges for software projects and agile development teams. To thoroughly investigate these challenges and propose workable solutions, this thesis employs a mixed-methods approach, utilising a systematic literature review (SLR) to understand the state-of-research, followed by a survey with practitioners to reflect on the state-of-practice.

In the SLR, we reviewed 45 studies to identify key data management aspects in those studies (including data integration, data collection, data quality, and data analysis). The results of the SLR identified several data management challenges, such as the complexity of automating data collection in dynamic environments, the difficulty of harmonising semantically diverse data, and the continuous struggle to maintain data quality standards throughout iterative development cycles. To address these challenges, the SLR reported various solutions from the reviewed studies, including utilising ontology-based data integration methods to tackle semantic inconsistencies, implementing automated quality assurance frameworks to improve data reliability, and adopting decentralised data management strategies to align better with agile practices.

The practitioner survey reported practical experiences from 32 agile practitioners across various industries, aiming to complement the findings from the SLR. The insights from the survey could enhance the practical application of our results and guide future research directions. The survey confirms the majority of SLR findings in terms of the data management challenges and solutions. However, the survey also offered additional practical insights, such as the need for better data management training, improved tools, and clearer communication in agile teams.

Based on these findings, this thesis presents implications for agile process activities (e.g., sprint planning and daily stand-ups), highlighting that inaccurate or lacking data during requirements gathering can result in poorly defined project goals and affect the entire development process. Furthermore, some of

the recommendations provided to help agile teams include the need for developing clear data management policies, training on data management tools, and adopting new data management strategies that enhance agility, improve product quality, and facilitate better project outcomes. This thesis encourages future researchers to explore new methods for data integration, real-time analytics, and data-driven decision-making in evolving agile practices.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Amjed Tahir, for his continuous support and invaluable guidance throughout this research. His research expertise has been crucial in ensuring that the outcomes of this work are not only valuable for future researchers but also validated and practical for software engineers in real-life scenarios across various industries. His insightful feedback and rigorous attention to detail have significantly strengthened the quality of this thesis, ensuring its contribution to advancing the field of agile software development and data management.

I am also deeply appreciative of Prof. Matthias Galster from the Department of Computer Science and Software Engineering at the University of Canterbury, New Zealand, and Prof. Peng Liang from the School of Computer Science at Wuhan University, China. Their valuable contributions to the review and development of this thesis, particularly through their involvement in the publication of our joint work, have significantly enhanced the quality and depth of the research.

Contents

List of Figures

List of Tables

List of Publications

This thesis is based on the following publication, which have been incorporated into the methodology, systematic literature review (SLR), survey, threats to validity and discussion chapters:

- Fawzy, A., Tahir, A., Galster, M., & Liang, P. (2024). Exploring Data Management Challenges and Solutions in Agile Software Development: A Literature Review and Practitioner Survey. Empirical Software Engineering (under review)

The manuscript is currently under review in the Empirical Software Engineering Journal (Springer). A preprint is made available on arVix [1].

Chapter 1

Introduction

Agile software development prioritises flexibility, rapid iteration, and constant adaptation to change, placing a high value on knowledgeable, capable teams. In the realm of business success, data is a crucial resource for making well-informed decisions and offers invaluable insights into customer behaviour and operational effectiveness [2]. Agile projects inherently produce diverse types of data, which need to be managed efficiently. Data Management in agile software development refers to the comprehensive process of collecting, storing, integrating, governing, and ensuring the quality and accessibility of data generated and used across various development cycles. This includes managing technical aspects like data integration, quality control, storage solutions, and analysis, as well as ensuring that data is secure, compliant, and aligned with the agile principles of flexibility and iterative improvement [3, 4, 5]. Agile projects face challenges in managing data across development data and product data. This thesis investigates these challenges and seek solutions from academic literature and practical specimens alike.

1.1 Problem Statement

The dynamic nature of agile methods can lead to issues related to managing this data (product, process, project, and operational data), such as integrating diverse data sources and maintaining data quality despite frequent updates and changes [6]. As software systems are developed quickly using agile methods,

keeping track of the latest and most accurate data can be challenging [7]. It is common for agile methodologies to emphasise working software rather than keeping detailed documentation. This can lead to missing or incomplete data records [4]. Software repositories, project management tools, user feedback systems, and other sources are common data sources used in agile projects. Integrating this data can be challenging and prone to errors [6]. Agile short development cycles (e.g., sprints) can result in rushed data collection, which raises the possibility of errors or omissions. Moreover, data collected by different agile teams often remained isolated in large software-intensive organisations, leading to inefficiencies and repeated efforts, as important feedback was not systematically shared across the organisation [8]. The data collected through agile processes can have different levels of detail, with some being summarised or grouped into broader categories instead of being recorded in fine detail. For example, instead of recording each specific user interaction, data might be aggregated to show the total number of interactions per day. This summarisation, known as data aggregation, can complicate data integration efforts by losing the finer details needed for specific analyses [9]. In order to achieve rapid delivery, agile projects frequently incur technical debt [10]. Workarounds or quick fixes to address that technical debt may result in data management issues (e.g., data quality issues [5]). Different agile teams might use different data management tools and procedures, which could lead to data being stored in a variety of formats and structures. The lack of standardisation makes integration and analysis tasks even more difficult. For instance, in a study on privacy requirements elicitation in agile development, it was found that 58.5% of practitioners do not use tools to document data privacy requirements, leading to inconsistencies and difficulties in maintaining standardised practices [11]. Furthermore, the dynamic nature of data management requirements (e.g., privacy requirements) and the frequent updates to user stories exacerbate the challenge, as teams often struggle to keep documentation current and comprehensive.

Those challenges related to managing data in agile software development can negatively impact the entire development process, potentially resulting in unmet stakeholder expectations, compromised project delivery, reduced agile team performance, and overall business success.

1.2 Research Questions and Objectives

Managing data in agile projects poses significant challenges that must be effectively handled to optimise organisational performance [5]. The main objective of this thesis is to systematically investigate data management challenges and potential solutions in agile software development. This thesis aims to provide a comprehensive understanding of these challenges and propose viable solutions to enhance data management efficiency in agile projects. Our thesis aims to answer the following two research questions (RQs):

RQ1. What are data management challenges in agile software development? This question aims to identify and categorise the various challenges that agile teams face in managing data.

Motivation: Understanding these challenges is crucial for improving data management practices within agile environments. By identifying the specific issues, the research aims to provide a foundation for developing targeted solutions that effectively address these challenges.

RQ2. What are the proposed solutions to address these challenges? This question aims to uncover the potential proposed or implemented solutions to address the identified data management challenges (RQ1). This could involve using specific tools, methodologies, frameworks, or practices that have been found effective in agile environments.

Motivation: This information is essential for practitioners to adopt and implement these solutions in their own projects, thereby enhancing data management practices and overall project outcomes.

1.3 Key Findings

We employed *a mixed-methods approach* using a systematic literature review (SLR) and a survey to explore data management challenges and solutions in agile software development. Combining the results from the SLR with the practitioner survey can ensure integrated knowledge from both academic and practical viewpoints. Moreover, the results from this thesis can inform future research directions and provide practical recommendations for improving data management practices in agile software development. *The thesis findings*, using the SLR, identified the key data management aspects and their challenges and

solutions as reported in previous studies. *The practitioner's survey* confirms the majority of the findings from the SLR and leads to new insights, such as the notable need for training programs that enhance data management skills within agile teams. Both the SLR and the survey show that agile teams usually struggle with managing data, such as integrating and collecting data from diverse sources, which hampers collaboration and well-informed decisions in projects. Several solutions have been proposed, including the use of ontologies, decentralised data management, automation tools, and communication-centric approaches. The practitioner survey emphasises the effectiveness of automation tools and decentralised data management practices in addressing data integration and quality challenges. The survey also emphasises how these challenges affect agile teams and how crucial it is to create concrete data management policies and offer team training. According to our research findings, managerial efforts should prioritise data management policy development, training, and tool adoption. Future research should focus on improved integration approaches, automated quality assurance, data collection, and real-time analytics.

1.4 Thesis Outline

This thesis structure is designed to offer a thorough investigation of agile software development's data management challenges and solutions.

1. *Chapter 2* provides foundational background knowledge of relevant literature on agile software development and data management. It demonstrates how important data management is in agile projects and points out the gaps that this thesis aims to fill.

2. *Chapter 3* describes our research methodology, including the SLR and the practitioner survey.

3. *Chapter 4* presents comprehensive results from the SLR by identifying the key data management aspects and their challenges and solutions found in published work.

4. *Chapter 5* presents the practitioner survey results that further investigate data management challenges and solutions.

5. *Chapter 6* provides an in-depth discussion of the results from both the SLR and the practitioner survey, exploring how these results can impact agile teams and project delivery, and the potential strategies those teams can follow to mitigate possible challenges.

6. *Chapter 7* discusses threats to validity and how they were reduced.

7. *Chapter 8* provides a conclusion of the thesis, considering contributions to industry practice and academic knowledge, and details practical implications and future research directions.

Chapter 2

Background

In this chapter, we cover the key areas of data management that this thesis builds upon. First, we discuss data management in agile software development, showing its importance in the agile context. We show how important data management is in agile projects and point out the gaps that this thesis aims to fill. We also discuss types of data in agile software development and their possible implications for agile processes.

2.1 Data Management in Agile Software Development

Agile practices have increased the effectiveness and responsiveness of software development to market demands [12]. Aiming to increase flexibility and efficiency, agile techniques first concentrated mostly on software development processes. However, as systems grew more complex and data-driven, it became clear that data management also needed to adopt agile principles. Often involving rigorous structures and significant upfront design, traditional data management techniques were progressively seen as incompatible with the agile approach, which favoured flexibility and ongoing improvement [13]. There is a cultural mismatch between the data management community, which favours traditional development approaches, and the agile development community, which emphasises collaborative techniques [4].

The shift towards distributed data management models, including data meshes,

has been one important change in this field. Unlike centralised and stable models usually used in data management, these models are made to support the ongoing updates and iterative nature of agile development [14]. This change reflects a larger trend towards data democratisation, in which data is made more accessible and controllable across many teams and departments, supporting more informed decision-making and more effective agile processes [15]

Data management encompasses all aspects of handling data, including its collection, integration, governance, and security [3]. It has grown in importance as agile methodologies have developed. Agile methods, by nature, include rapid iterations and short feedback cycles, requiring effective data management to inform decision-making processes [7]. However, there are multiple data management challenges in agile software development. For example, handling unstructured data, which is prevalent in big data environments, is one of the data management challenges. Unstructured data makes the analysis and integration of data difficult [16]. Another challenge in agile software development is the frequent data changes, especially in big data scenarios. Agile teams must constantly adjust to fit this dynamic nature of data, which can be difficult to properly control [7]. Ensuring data quality and governance in agile processes presents another difficult task. Agile methods give speed and flexibility a top priority, which occasionally runs counter to the strict standards for data quality and governance [11]. Despite these challenges, a comprehensive understanding of data management challenges and their potential solutions has not been explored in depth.

2.2 Previous Work

Previous research established the foundation for understanding data management in agile software development. For example, Graetsch et al. [5] discussed the difficulties of dealing with data challenges when delivering data-intensive software solutions. Their research highlighted the vital role effective data management plays in making sure software projects are successful. Similarly, the solution proposed by Rosenkranz et al. [17] addresses specific data integration (e.g., harmonising data from different sources) and data quality challenges (e.g., ensuring accuracy and consistency), highlighting the interconnected nature of these two aspects in effective data management. Their contributions provided a

deep understanding of the fundamental challenges in data management for agile software development.

The cultural mismatch between agile development communities and data management was revealed in a study by Ambler [4], underscoring the need for more integrated and cooperative approaches. Effective data management in agile projects may be hampered by the major differences in practices and priorities that frequently arise from this cultural mismatch. For instance, Ambler discovered that although nearly 62% of IT organisations encountered problems with production data, 18% did not have a plan in place to deal with these issues. Additionally, roughly 25% of respondents said that renaming a column in a production database could take longer than a day, with some estimating it would take up to three months or thinking it was too dangerous to try. These differences in data management techniques point to a larger cultural gap in which agile teams prefer more evolutionary, iterative methods while data professionals frequently follow traditional, serial approaches. According to Ambler's survey, 19% of development teams felt their data teams provided too little value, and 36% of development teams felt their data teams were too slow to work with. These findings highlight the need for more efficient integration between these communities in order to enhance data quality and project outcomes.

Recent research studies have continued to investigate new challenges and solutions in data management in agile software development. For instance, techniques for collecting diverse data using user-centered design have been proposed as important solutions to data management challenges [18]. This technique highlights how important it is to incorporate iterative testing and user feedback into the data collection process. Other recent studies have focused on the necessity of real-time data utilisation and integration in agile settings. Research conducted by Matthies et al. [19], Barbala et al. [15], and Lin et al. [20] have demonstrated the importance of efficient data collection and use to support decision-making and quickly adjust to changes. Other research has built upon these foundational works by addressing more specific challenges. For example, the difficulty in accessing data collected by different agile team members during various development phases has been highlighted as a significant issue [8]. This fragmentation can lead to data loss and inefficiencies during critical project handovers. Moreover, the lack of concurrent access to data, as discussed by Kaur [21], hinders real-time agile team cooperation and knowledge exchange,

which are essential for agile methodologies. Legal and compliance issues are also raised by the uncertainty surrounding the data that agile product teams are allowed to collect, as mentioned by Barbala et al. [15]. This thesis builds upon the foundational and recent research identified through the SLR. By employing a mixed-methods approach, the thesis aims to provide a more comprehensive understanding of data management challenges and solutions in agile software development.

2.3 Types of Data in Agile Software Development

Organisational success is significantly impacted by managing data, which is a crucial resource that helps with decision-making, provides insightful information about customer behaviour, and improves operational efficiency [2]. Many forms of data (e.g., business or product data) play important roles in agile development, each impacting the development process and the project's overall success. Different data types can have different implications for agile processes. We discuss those data types and their possible implications below:

Business data: This data type contains details relevant to business decisions, such as information for forecasting sales, anticipating the need for raw materials, and analysing consumer behaviour [2]. This data type enhances productivity and operational efficiency by facilitating direct feedback from users and enabling quick communication between developers and customers in an agile process. This data facilitates iterative cycles of decision-making in agile processes. Teams can use it to prioritise features according to market trends and customer requirements, ensuring that development efforts align with company objectives [8].

Product Data: This data type contains details about the software product itself, including design documents and source code. It measures the software product's size, complexity, and design structure—all of which significantly impact software quality [22, 23, 24]. This data provides metrics that can be used to measure code quality, complexity, and technical debt to inform continuous integration and agile delivery processes. Such metrics are critical to sustaining high levels of software quality and enabling quick releases [17].

Process Data: Process data is about the software development process and

includes several elements that could affect how much work is needed to build a software system. Understanding and enhancing the software development lifecycle need the use of process data [22, 23]. Process data is essential for continual improvement, which agile techniques emphasise. Teams can modify their procedures for improved results by using the insights it offers about process bottlenecks, team performance, and collaboration efficiency [2].

Project Data: This data type focuses on a software project's general health and state, including resources, risks, budgets, and schedules. Project managers frequently use this data to track the status of their work and inform choices about planning and controlling their projects [24]. Agile project management depends on having real-time project data. It assists in monitoring development, controlling risks, and making sure the project stays within budget and on schedule. This data demonstrates the agile principles of transparency and ongoing feedback [2].

Operational data is data about a company's daily activities. It entails privacy engineering, managing personal data, and deploying and running software in cloud environments [25]. Operational data comprises sensitive corporate information, classified documents, and data kept in databases necessary for enterprise operations. Such data is crucial to the day-to-day functioning of the business [26]. Within the agile framework, operational data facilitates the daily management of projects.

These different data types show how they may be used for better decision-making, encourage client participation, and enhance software development procedures—all of which contribute to an organisation's success. Agile teams can make more informed decisions when they better understand the various forms of data and their functions. For example, combining *process* and *business* data can reveal detailed information about customer satisfaction and project performance [8]. Categorising data makes it easier to pinpoint the unique challenges of each type of data. For instance, monitoring operational data is crucial for compliance and security, while guaranteeing the correctness and consistency of product data is crucial for preserving software quality [2]. Developing targeted solutions for specific challenges is made more accessible by classifying data. When the type of data and its challenges are well-defined, for instance, applying ontologies for data integration or using automated methods for data quality assurance can be done more successfully [27].

2.4 Summary

This chapter provides background about agile methodologies and the significance of data management in agile software development. The chapter emphasises how important data management is to assure data availability, dependability, and accessibility in a timely manner, supporting agile approaches with their rapid feedback cycles and short iterations. The chapter also presents and classifies various data types that can be used as part of agile products. Built on the background knowledge presented in this chapter, the next chapter 3 will detail a mixed-methods approach employed to investigate data management challenges and potential solutions, ensuring a robust and reliable foundation for our research findings.

Chapter 3

Methodology

In this chapter, we describe the research methodology followed in this thesis. The research follows a mixed-method approach using a systematic literature review (SLR) and a practitioner survey (Section 3.1 and Section 3.2). This chapter is structured to provide detailed information about the research design, data collection, and analysis processes employed in the thesis. The SLR methodology explains the methodology for the SLR, which was employed to gather, organise, and analyse existing research on data management challenges and solutions in agile software development. The survey methodology details the design and execution of the practitioner survey. It also discusses the qualitative and quantitative analysis techniques applied to analyse the survey data.

3.1 SLR Methodology

As noted in Chapter 1, the first step of our investigation was to conduct an SLR of data management challenges and solutions in agile software development. The review aimed to answer the two RQs: (RQ1) What are data management challenges in agile software development? and (RQ2) what are the proposed solutions to address these challenges? The SLR was designed following Kitchenham and Charters [28] guidelines. It drew upon works related to other literature reviews for agile methodologies, such as Dikert et al. [29] and Campanelli and Parreiras [30], by adopting their systematic approaches to reviewing the literature. The review was composed of several stages, as demonstrated in Figure

3.1. Details of each stage are presented in the following sections.

3.1.1 Search Process

We used Scopus to search for relevant studies due to its comprehensive indexing of major publications in software engineering, as highlighted in studies by Mourão et al. and Carrera-Rivera et al. [31, 32]. The extensive coverage of Scopus (it covers a wider range of literature in closely related fields) provides access to high-quality, peer-reviewed literature, which is crucial for an in-depth exploration of agile software development. Moreover, data management is a topic at the intersection of multiple disciplines, including software engineering, project management, IT management, data science, and information systems. Scopus provides more extensive coverage of literature on our study topic as it covers popular publishing venues, including the four major software engineering publication libraries: IEEE Xplore, ACM Digital Library, SpringerLink and ScienceDirect. The search covered literature up until October 2023, the date we conducted this review.

We constructed a search string covering the main keywords of the studies we aim to review. We first identified a set of initial keywords for our search string: *data management, agile, challenges and solutions*. This set of keywords was iteratively refined to review article coverage. We experimented with different search strings until we reached a string that returned relevant results. We had a quasi-gold standard [33] as a set of studies we knew were relevant and the search string found them all (included in our replication package [34]).

Before finalising our search keywords and search string and establishing our inclusion and exclusion criteria, we initiated a pilot search with a basic string ("agile AND data") to search for material in the Scopus database. This initial search generated 9,900 studies, yet a review of a sample of titles and abstracts revealed their lack of relevance to our focus on data management challenges in agile software development, being too broad for our purposes.

To refine our search and cover more pertinent results, we limited the search string by incorporating the term "challenge" and adding synonyms like "obstacle", "issue", and "problem" with the OR operator. This refinement reduced the number of results by a third to 3,068 studies. However, a further examination of a new sample of titles and abstracts showed that they were not all explicitly related to software development, as our research specifically targeted agile soft-

ware development, not agile methodologies in a broader sense. We continued to refine our search string, modifying the search string to "(agile AND data AND software) AND (challenge OR practice)" which led to 1,396 studies. Yet, these results still did not focus explicitly on data management. To address this issue, we further refined the search to "(agile AND data AND software) AND (challenge OR practice) AND ("data management")", achieving more targeted results. However, this specific approach risked missing relevant studies where "data management" appeared in the context of its various aspects rather than as an explicit term. To overcome this, we expanded the search string to include a range of terms associated with data management, resulting in a more balanced set of 181 studies as of the final execution in October 2023. To avoid missing related studies by focusing narrowly on specific terms, we used Boolean operators (OR) among them or included the general term "data management". A review of a random sample of 30 titles and abstracts from these 181 studies shows greater relevance to our research question.

As illustrated in Figure 3.1, the search proceeded with the 181 selected studies. Next, we conducted a full pilot review on a sample of these 181 studies. The purpose of this pilot review was to ensure the relevance and alignment of the selected studies with the research objectives before proceeding to a full systematic review. It also helped to refine the inclusion and exclusion criteria and assess the initial data extraction process to ensure the quality and reliability of the final analysis. We applied the inclusion and exclusion criteria, performed quality assessments (see Section 3.1.5) and extracted the data to ensure the final selection aligned with our research objectives.

In the search string, the goal was to cover all two aspects of the study, i.e., *agile software development* and *data management*, with the goal of capturing studies that focus on data management (aspects, issues, challenges and solutions) in agile software development. Our final search string (which covered the aspects above) is shown below:

```
(agile AND software AND data) AND (challenge OR practice)
AND ("data quality" OR "data management" OR "data
security" OR "data governance" OR "data integration"
OR "data storage" OR "data privacy" OR "data access" OR
"data analytics" OR "data validation" OR "data capture"
OR "data modeling" OR "data virtualization" OR "data
cataloging" OR "data Versioning" OR "data monitoring" OR
"data transformation" OR "data archiving" OR "backup" OR
"disaster recovery" OR "data-driven")
```

3.1.2 Inclusion and Exclusion Criteria

We selected studies that met all of the following inclusion criteria:

1. Studies that discussed at least one aspect of data management in the context of agile software development (including challenges and solutions).
2. Studies that were fully accessible and available in full text for comprehensive analysis.
3. Studies that were published in English.

We applied the following exclusion criteria and excluded studies when they met one of these criteria:

1. Studies that did not specifically mention agile software development or agile methodologies.
2. Duplicate studies to ensure the uniqueness of each data point.
3. Studies that focused exclusively on technical or other aspects of software development without demonstrating a clear link to data management challenges in agile software development.
4. Non-research materials such as tutorials, books or workshop summaries.
5. Opinion pieces, editorials, or non-peer-reviewed articles to ensure the academic rigour of the review.

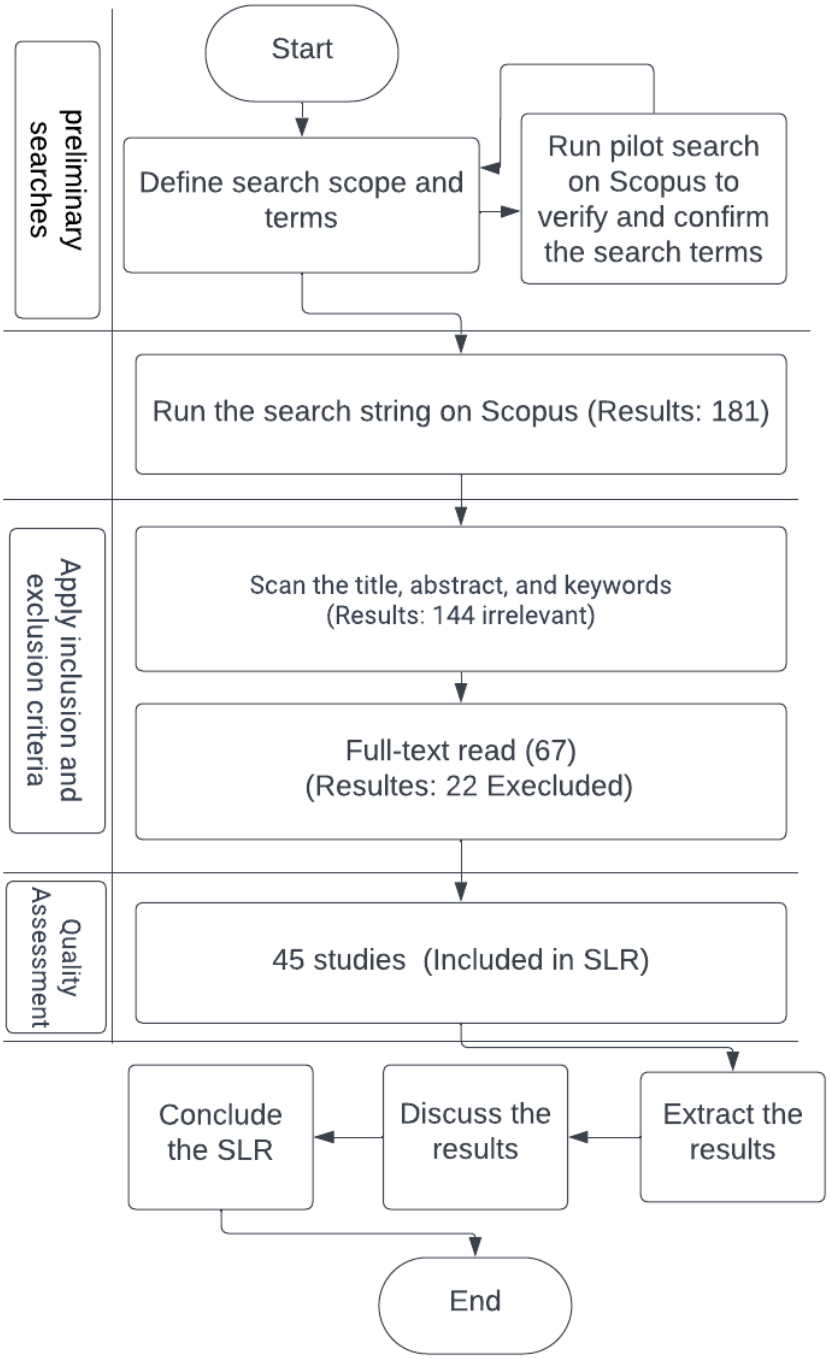

Figure 3.1: The Review Process

3.1.3 Study Selection

We screened all of the 181 studies, and we carefully checked each study's title, abstract, and list of keywords. After applying the inclusion and exclusion criteria, we filtered out 114 irrelevant studies, leaving 67 relevant studies. The irrelevant studies were primarily excluded because they either did not focus on data management within agile software development or were non-research materials lacking the required academic rigour.

To ensure that these studies precisely matched our research questions and scope, we conducted a full-text review, applying the same criteria. This review led to the exclusion of 22 studies from the initial 67. Reasons for exclusion included factors such as language (e.g., one study had only the abstract written in English), content type (e.g., one was a tutorial, another an event), or lack of depth or relevance to our scope. Consequently, 45 studies were finally included.

3.1.4 Data Extraction

We utilised a combination of thematic analysis and content analysis to synthesise data from the studies. We conducted a thematic analysis [35, 36] to identify, analyse, and report themes (on data management aspects, challenges, and solutions) within the qualitative data extracted from the 45 studies. Initially, a subset of studies (a quasi-gold standard of nine studies) were reviewed to identify general data management aspects. This provides a structured understanding of the various aspects of data management within agile software development. Additionally, this foundational knowledge helps identify specific challenges and solutions associated with each aspect, thereby addressing our two RQs.

We categorised the extracted data from the studies (all those items were extracted as free text) as follows:

- study objective
- data management aspect covered
- challenges identified (RQ1)
- suggested/implemented solutions (RQ2)

Subsequently, those aspects were discussed in detail with at least one additional reviewer (a collaborator) to ensure coverage of all challenges and solutions. Relevant data was retrieved and organised by careful iterative reading through the complete texts of 45 included studies. Challenges were identified based on descriptions of problems faced in managing data within agile projects, as documented in the reviewed studies. Solutions were extracted by noting specific strategies, tools, or methods proposed or implemented to address the identified challenges. For example, the thematic analysis revealed five challenges associated with data integration, such as data harmonisation, semantic heterogeneity, data transformation, and extraction. The identified solutions were also thematically categorised and presented. We iteratively reviewed and validated the extracted data. To guarantee accuracy and consistency, we discussed the extracted data from a sample of studies and compared the outcomes. To improve clarity and comprehensiveness, the outcomes were discussed with at least one additional reviewer (a collaborator) and settled collectively, and the extraction forms were modified as a result.

We conducted a content analysis of the selected studies to classify them based on the data management aspects they addressed (such as data integration, data quality, data collection, and data analysis). Additionally, we quantified the number of studies discussing each aspect to identify the areas that covered the most and least. For example, the studies were classified into 15 data management aspects (see Section 4.1), listed these aspects, and showed the distribution of studies across these aspects, providing a clear quantification of how many studies addressed each aspect. In addition, the data types discussed in the studies are provided in Appendix C.

We provide a dataset of all the included studies we have extracted and our detailed analysis in an external replication package [34].

3.1.5 Quality Assessment

Throughout our review, we have conducted comprehensive quality checks to ensure rigour, integrity, and relevance. This process began with examining selected samples of search string results, aligning them carefully with our established inclusion and exclusion criteria to ensure the focus and relevance of our study. Furthermore, we conducted an in-depth quality assessment of the 45 studies, employing a streamlined, three-point scoring system across six key criteria [37]:

Clarity of Objectives, *Appropriateness of Methodology*, *Adequacy of Data Analysis*, *Relevance to Research Questions*, *Rigor of Data Collection*, and *Quality of Reporting*. Each criterion is rated 1 (adequate), 2 (good), or 3 (excellent), depending on the study's clarity, methodological soundness, analytical rigour, relevance, data collection thoroughness, and quality of reporting. We calculate the total score for each study by summing the scores from all criteria, with the maximum possible score being 18. The studies are then categorised according to their total scores, with 7-11 indicating adequate quality, 12-15 indicating good quality, and 16-18 representing excellent quality. This meticulous approach ensures a comprehensive and nuanced assessment of each study's quality, aligning with our goal to exclude low-quality research and thereby maintaining the integrity and reliability of our findings. The quality assessment confirms that the 45 studies are of good quality (the scores of these 45 studies are all above 12). Details of the quality assessment, including the scoring system, can be found in the replication package [34].

3.2 Practitioner Survey Methodology

Based on the challenges and solutions reported in the literature, the survey aimed to better understand what practitioners identified as challenges (RQ1) and solutions (RQ2). Furthermore, insights from agile teams across different industries could enhance the relevance of the findings from the SLR for the practical application of our results and for guiding future research directions. We have followed the guidelines of Punter et al. [38] and Linåker et al. [39] in designing and conducting our practitioner survey. The survey was composed of several stages. Figure 3.2 shows the overall survey process. Details of each stage are presented in the following sections.

3.2.1 Survey Design

The SLR informed the survey by guiding the development of specific sections based on data management aspects, ensuring comprehensive coverage of pertinent challenges (RQ1) and solutions (RQ2). For example, our analysis of the SLR emphasised several data management challenges, such as data harmonisation and semantic heterogeneity in data integration. This has led to the addition of specific questions to the survey on specific integration challenges

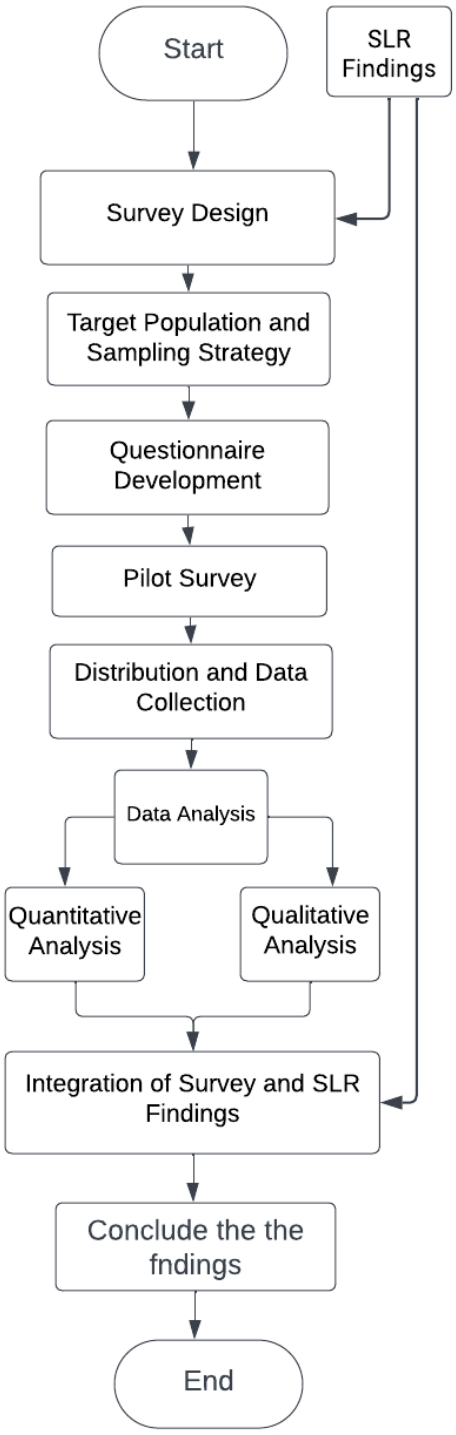

Figure 3.2: The Practitioner Survey Process

and solutions. The SLR also identified several challenges in the literature, and the survey addresses them by asking for detailed practitioner experiences and solutions. For example, in the survey, we directly asked practitioners about the impact of these challenges on their role and the project delivery process.

Target Population and Sampling Strategy

Our target population included software practitioners across various roles and experience levels who also worked in agile development projects. This encompassed senior developers, team leads, scrum masters, product owners, analysts, project managers, quality managers, DevOps engineers, data engineers, and solution architects. We employed a purposive sampling strategy [40] aimed at selecting individuals who were experienced in agile software development and had practical insights into data management challenges and solutions. This approach ensured that the data collected was rich in detail and relevant to our research questions.

The survey was fully conducted online using Microsoft Forms. The survey link was circulated to practitioners through personal connections and local networks. We aimed to distribute the survey to the target population, specifically those strongly interested in data management. The survey was distributed internationally to agile practitioners through direct contact at different organisations, through LinkedIn, and emails. The survey was distributed in March 2024, and the response period was open until June 2024. Responses were closely monitored to ensure the progress of the data collection process. To boost responses, follow-up reminders were sent to the contacts list to encourage participation.

Questionnaire Development

The questionnaire was first reviewed by an experienced researcher in survey studies who was also not involved with the study. Based on the experienced researcher's feedback, we refined the survey questions for clarity, ensuring they were more closely aligned with the study's objectives. Specifically, the researcher suggested rephrasing certain questions to avoid ambiguity and adding additional options in the multiple-choice sections to better capture the range of possible responses.

Participants could take the survey anonymously and choose not to provide information that could identify them. All data from the survey were securely

stored online. The survey questions could be accessed online [1] and also available in Appendix D. The survey was reviewed and approved by Massey University's ethics committee on February 27, 2024 (Reference Number 4000028568). The survey instrument (questionnaire) is divided into the following sections:

1. *Purpose and consent:* Participants were given information on the survey's objective and consent information (this was mandatory to accept).
2. *Introduction:* This section explained the relevance of data management in agile projects.
3. *Background:* This section collected demographic information from participants, including age, gender, professional seniority, years of experience, educational background and role in agile projects, and industry sector (pre-defined sectors with an option to add more). Additionally, we asked a mandatory question to confirm if the participant had worked on projects that followed agile software development practices. Participants are allowed to continue only if they confirmed that they did.
4. *Instruction:* This section provided instructions on how to answer the survey questions. Practitioners were advised to establish their answers based on their experience with agile projects.
5. *Data management challenges and solutions:* This section collected answers related to (data integration, collection, quality, and analysis) challenges and solutions based on the findings from the SLR, but also allowed participants to comment on their answers in an open-ended field. We employed a mix of multiple-choice and open-ended questions for each data management aspect section. For example, in the data integration section, we asked open-ended questions about different aspects of data integration, such as "*What data integration challenges have you encountered in your projects?*", followed by multiple-choice to ask the same question with giving choices. The same style of questions (open-ended followed by multiple-choice) has been employed for each data management aspect section.
6. *Recommendations:* This section asked the participants to provide information about the impact of data management challenges on their respective roles, project delivery, and potential enhancements.

Before deploying our survey, we conducted a pilot run with three practitioners with more than ten years of experience in agile and data management (senior

[1] `https://forms.office.com/r/5b6vTcSjHV`

agile project manager, senior data analyst, and senior developer). The pilot aimed to verify the questions, scope, and estimated completion time. Feedback from the pilot was used to refine the questionnaire structure, clarify the questions, and address any identified validity threats. For example, we rephrased some questions for clarity and made structural changes to the survey flow for better readability. Additional questions were included based on the recommendations of those who have taken the pilot survey.

3.2.2 Data Analysis

We conducted a quantitative and qualitative analysis to analyse the survey data we obtained from all participants.

Quantitative Analysis:

We first gathered statistics from the responses, including demographic information and response frequencies. We then investigated the general trends and patterns in the collected data. The demographic data helped us understand the variety of participants in our survey, such as their job roles, experience levels, and industry sectors. It was important to make sure our findings were based on a broad range of perspectives from the target population and not just from one specific group. The response frequencies showed us how common certain answers were among the participants. This helped us identify which challenges and solutions were most frequently mentioned. By looking at these numbers, we could see what issues were most widespread and which solutions were commonly used. Including both demographic information and response frequencies in our analysis allowed us to better understand the results. It ensured that our conclusions were relevant to different groups within the software development community and provided a solid base for further analysis.

Qualitative Analysis:

We used thematic analysis to analyse the qualitative data, following the guidelines established by Braun and Clarke's thematic analysis [36]. This analysis employed a combined approach, using deductive, inductive and semantic methods.

- *Deductive*: We used a deductive approach by starting with predefined categories we already established from our SLR findings. For example, we asked questions like "*What data integration challenges have you encountered in your projects?*" and "*What solutions have you employed to address data quality challenges?*" to create the predefined deductive categories such as data integration challenges and data quality solutions.

- *Inductive*: We also used an inductive approach, where we let new themes come up naturally as we analysed the responses. For instance, the theme "*Data Structure and Format Challenges*" came naturally from the codes (participants' responses) "*Inconsistent Data Structures*" and "*different Data Formats*". This new theme was then categorised under the predefined deductive category of data integration challenges.

- *Semantic*: Our analysis was semantic, meaning we focused on the clear, straightforward meanings of the participants' responses. For example, when a participant mentioned "*We faced issues with data consistency across different datasets*", we simply coded it as a "*Data Consistency Issue*" without interpreting deeper underlying causes.

We carried out the qualitative analysis using NVivo (version 14)[2]. The steps taken for the thematic analysis are explained below:

- *Data Familiarising*: we first familiarised ourselves with the data by reading over the survey responses. Iterative rereading of the responses helped to identify initial themes and codes. At each step, we took notes on possible codes and themes.

- *Developing Initial Codes*: We created the initial codes by systematically analysing responses to each open-ended question. This process was conducted question by question, where we carefully examined each response and identified recurring themes, concepts, or keywords that were relevant to the predefined deductive category. These themes were then categorised into initial codes. To ensure consistency and reliability in the coding process, another reviewer (a collaborator) independently reviewed the developed codes. The reviewer provided feedback, and any discrepancies

[2] `https://lumivero.com/products/nvivo/`

were discussed and resolved to achieve consensus on the final set of initial codes.

- *Identifying Themes*: codes were categorised into potential themes. Those themes were examined for consistency and coherence. Each theme was given a descriptive name that appropriately reflected its content. We also identified sub-themes as needed. This was done with another reviewer (a collaborator) as we iteratively discussed and refined the identified themes. For instance, the codes "*Inconsistent Data Structures*" and "*different Data Formats*" were categorised under "*Data Structure and Format Challenges*".

- *Reviewing Themes*: To strengthen validity, we reviewed the themes while another reviewer (collaborator) confirmed them. We conducted two review phases:

    1. Review 1: reviewing themes in light of participants' quotes under the theme: we checked the quotes from participants to make sure they accurately represented that theme.
    2. Review 2: reviewing themes in light of the entire dataset: We looked at the themes in the context of all the survey responses by checking how well the themes represented the entire set of responses and making adjustments if needed to ensure they accurately reflected the data and our RQs.

- *Comparison with SLR-Based Options*: we compared the identified themes from the open-ended question with the options provided in the multiple-choice question. This helped integrate the findings into the results section.

3.3 Summary

This chapter describes the research design used to explore data management challenges and solutions in agile software development. It used a mixed-methods approach, combining an SLR and a practitioner survey. To find prevalent challenges and suggested solutions in the literature, the SLR entailed a thorough search and analysis of 45 existing studies. The practitioner survey collected data from 32 professionals in agile software development and data management to

capture practical insights and validate the findings from the SLR. The SLR was followed by a practitioner survey. The survey included demographic questions and sections on the key data management aspects (data integration, collection, quality, and analysis), addressing both challenges and solutions. Overall, this chapter provides a framework for systematically investigating and validating data management challenges and solutions in agile software development, aiming to fill the identified gap in the literature. The SLR results formed a foundation for understanding the key aspects of data management and the associated challenges and solutions, establishing the stage for the detailed analysis and discussion of the survey results. Next, we present the SLR results in Chapter 4, followed by the practitioner survey results in Chapter 5.

Chapter 4

Systematic Literature Review on Data Management in Agile Software Development

This chapter provides a detailed review and analysis of various aspects of data management as presented in the original studies we surveyed. To address the research questions posed at the outset of this thesis (RQ1: What are the data management challenges in agile software development? and RQ2: What are the proposed solutions to address these challenges?), we start by defining various data management aspects in Section 4.1. Based on these aspects, we then discuss the data management challenges (Section 4.2) and solutions (Section 4.5) associated with each aspect in detail. Note that we discuss the challenges and solutions together, as this will provide readers with a better context of each challenge and the related solution as discussed in the related study.

4.1 Data Management Aspects

Data management aspects refer to the different key areas involved in handling and using data effectively in agile software development. These areas include tasks like integrating data from different sources, ensuring data quality, collecting data, and analysing it. Each aspect addresses a specific challenge that must

be managed to ensure smooth and effective agile development.

We classified each of the 45 studies based on the data management aspects they address. To do this, we carefully reviewed each study to identify the specific areas of data management they focused on, such as data integration, quality, collection, and analysis. This helped us group the studies under common themes and draw meaningful conclusions. Table 4.1 lists all 15 data management aspects across studies. Figure 4.1 shows the distribution of studies across the 15 data management aspects we identified.

4.1.1 Data Management Aspects Definitions

Below, we present the definitions of the data management aspects that we extracted from the included studies in the SLR. Table 4.1 lists the studies we analysed and the aspects they address. This table will serve as a reference as we define each aspect below.

- *Data Integration*: The process of combining data from various sources and systems into a unified and consistent view. This consolidated data is then utilised for analysis, reporting, and decision-making purposes.
- *Data Collection*: The activities involve gathering and distributing information. This information is subsequently utilised for analysis and decision-making.
- *Data Quality*: The accuracy, completeness, and consistency of a dataset. The importance of high-quality data is underscored by its critical role in effective data analysis and decision-making processes.
- *Data Analysis*: The comprehensive activity of inspecting, cleaning, transforming, and modelling data to unearth valuable information, obtain conclusions, and strengthen decision-making. This activity necessitates the management of diverse data types, including structured and unstructured data, and requires diverse techniques and tools.
- *Data Storage*: The process of saving data in a digital format, which can be retrieved and used later.
- *Data Validation and Governance*: The process of ensuring data accuracy, consistency, and security throughout its lifecycle.

- *Data Ingestion*: The process of obtaining and importing data for immediate use or storage in a database.
- *Data visualisation*: The graphical representation of information and data, making complex data more accessible and understandable.
- *Data-Driven Decision Making (DDDM)*: The practice of making decisions based on data analysis rather than intuition or observation alone.
- *Data Privacy*: The proper handling, processing, storage, and usage of personal information, ensuring compliance with legal requirements.
- *Data Security and Compliance*: The process of protecting data from unauthorised access or corruption, while compliance refers to adhering to laws and regulations related to data.
- *Data Testing*: The process of ensuring the accuracy, completeness, and reliability of data in databases and data-intensive applications.
- *Data Management Development*: The process of developing, executing, and supervising plans, policies, programs, and practices that control, protect, deliver, and enhance the value of data and information assets.
- *Data-Driven Development (DDD)*: The focus is on the data itself as the main driver of development rather than traditional software development approaches.
- *Data Maintenance Strategy*: The ongoing process of keeping data up-to-date and ensuring its quality and accuracy over time.

4.1.2 Overlapping Data Management Aspects

The various data management aspects of agile software development are closely connected, as demonstrated in the studies shown in Figure 4.2. For example, the studies show that data integration and data quality are closely linked because, when combining data from different sources, it's essential to ensure that the data remains accurate and consistent. Furthermore, inadequate integration can result in quality problems, making the data unreliable for analysis. Similarly, the studies show that data collection impacts data integration. The proper data

Table 4.1: Classifications of Studies Across Data Management Aspects

	Data Management Aspect	**Ref.**																		
1	Data Integration	[16]	[41]	[42]	[43]	[44]	[45]	[27]	[46]	[14]	[47]	[48]	[49]	[17]	[50]	[51]	[52]	[53]	[54]	
2	Data Storage	[16]	[41]	[15]	[27]	[51]	[52]	[55]	[53]											
3	Data Validation and Governance	[16]	[21]	[56]	[49]	[13]														
4	Data Quality	[6]	[18]	[57]	[41]	[9]	[4]	[56]	[58]	[47]	[48]	[49]	[17]	[50]						
5	Data Ingestion	[16]	[42]	[59]	[60]	[58]	[55]													
6	Data Collection	[18]	[2]	[57]	[43]	[42]	[8]	[44]	[60]	[9]	[15]	[21]	[19]	[56]	[61]	[58]	[52]	[20]	[55]	[62]
7	Data Security and Compliance	[44]	[26]																	
8	Data Privacy	[11]	[25]	[14]	[15]	[61]	[51]													
9	Data Analysis	[16]	[41]	[42]	[2]	[57]	[43]	[60]	[9]	[19]	[61]	[27]	[58]	[46]	[52]	[20]	[53]	[63]	[62]	[8]
10	Data Visualisation	[41]	[57]	[56]	[46]	[48]	[20]	[63]	[62]	[58]										
11	Data-Driven Decision Making	[18]	[42]	[2]	[57]	[27]	[64]	[20]												
12	Data-Driven Development	[15]	[65]	[61]	[53]															
13	Data Testing:	[44]	[9]	[4]	[50]															
14	Data maintenance strategy	[4]	[14]																	
15	Data Management Development	[13]	[66]	[67]	[53]	[16]	[43]													

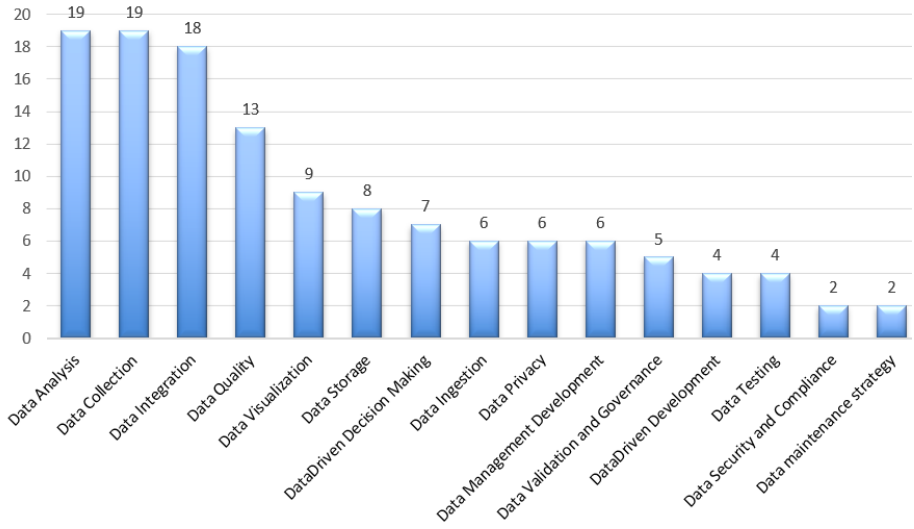

Figure 4.1: Total Number of Studies Discussing Challenges and Solutions Associated with the Data Management Aspects

collection methods help ensure that data is organised and ready for integration, reducing extra work. Data analysis also depends heavily on how well data is integrated and the quality of that data. Effective integration and high-quality data are crucial for drawing accurate insights during analysis. On the other hand, the needs of analysis can influence how data is collected and integrated to meet specific goals. By noticing these connections, we can better manage data in agile projects and ensure that addressing challenges in one aspect strengthens other aspects as well.

Before delving into the specific challenges and solutions, ***we will focus on the four key aspects of data management*** that were identified as the most significant in our SLR. The selection of these aspects (data integration, data collection, data quality, and data analysis) was based on their frequent discussion and significance in the studies we reviewed (see Figure 4.1). Each aspect represents crucial areas where challenges can significantly impact data management efficiency in agile software development. These aspects are essential for ensuring that data is effectively managed, integrated, and utilised throughout the agile development process. In the upcoming sections, we will explore the specific challenges associated with each aspect and present the solutions suggested in

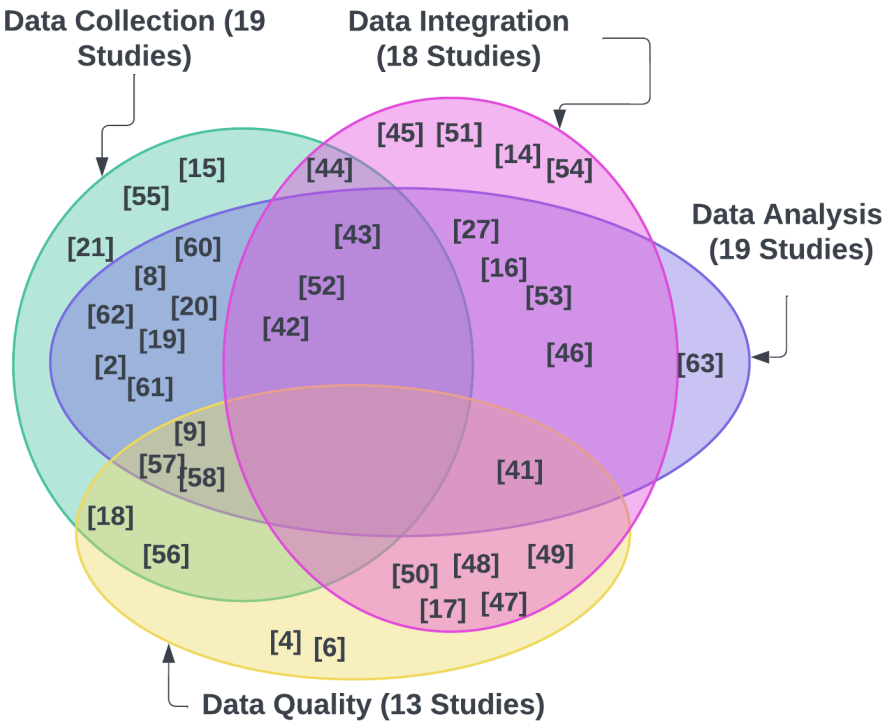

Figure 4.2: Overlapping Studies Between the Key Focus Aspects in Data Management

the literature to tackle them.

4.2 Data Integration Challenges and Solutions

We have identified several challenges associated with data integration in agile software development. These challenges and their proposed solutions are grouped into categories, which we discuss below. Furthermore, a summary of these challenges and their corresponding solutions is presented in Table 4.2.

Table 4.2: Summary of Data Integration Challenges and Solutions

Data Integration Challenge	Data Integration Solution	Solution Status	Ref.
Data Harmonisation and Interoperability	Development of ontologies	Proposed	[45]
	Cloud-based platform with agile data-loading pipeline	Implemented	[47]
	Agile workflow for integrating data	Proposed	[54]
	Translating the metadata of the datasets into different formats	Implemented	[51]
Semantic Heterogeneity	Communication-centric approach with agile methods	Implemented	[17]
	Development of ontology-based approach	Proposed	[42]
	Development of domain ontologies	Implemented	[27]
	Implementing a modular and agile framework	Implemented	[46]
Data Transformation and Extraction	The implementation of advanced ETL procedures	Implemented	[49]
	Data visualisation and federation layer	Proposed	[53]
	Replicable ETL process	Implemented	[48]
	Shift to advanced analytic platforms	Proposed	[52]
Managing Data Integration	Adoption of 'data mesh' approach	Proposed	[14]
	Adoption of the Microsoft Solutions Framework (MSF)	Implemented	[50]
	Architecture-centric approach with AABA methodology	Implemented	[16]
Diverse and Decentralised Data Sources	Developing a common product line architecture	Implemented	[41]
	Automated Continuous Quality (ACQ) Metrics dashboard	Partially implemented	[43]
	Solution referenced in Section 4.3	Partially implemented	[44]

Data Harmonisation and Interoperability: Abdallah and Fan [45] discussed the challenge of harmonisation and interoperability between heterogeneous Maintenance Repair and Overhaul (MRO) information systems, which is a critical issue in integrating maintenance records over the lifespan of an aircraft. Spengler et al. [47] discussed the challenge of integrating data from distributed and heterogeneous sources at the technical, structural, and semantic levels. Vøgt et al. [54] addressed incorporating new climate data into local government processes. Schüttler et al. [51] focused on coordinating the definition and integration of datasets to ensure interoperability and avoid the development of parallel structures, necessitating alignment with other research infrastructures like the European Biobanking and BioMolecular Resources Research Infrastructure - European Research Infrastructure Consortium (BBMRI-ERIC)

and the German Medical Informatics Initiative (MII), which is a national initiative aimed at improving medical research and healthcare by facilitating the sharing and integration of data across various medical institutions.

To address the data harmonisation and interoperability challenges Abdallah and Fan [45] proposed a solution, which is the development of ontologies to semantically integrate heterogeneous maintenance records of MRO systems, enabling consistent understanding and data exchange across different systems. This approach, known as Ontology-Based Data Integration (ODBI), is supported by the Agile Development for Ontology-Based Applications (ADOBA) methodology, which combines ontology development with application integration in an agile framework. The solution was conceptualised and detailed within the framework of the ADOBA methodology. However, Abdallah and Fan [45] indicated that further validation is necessary to assess the accuracy and applicability of the methodology in developing ontology-based applications. Spengler et al. [47] presented a cloud-based platform as a solution that includes a generic agile data-loading pipeline. This pipeline supports deploying and customising clinical and translational data warehousing solutions, specifically i2b2 and tranSMART. It addresses the need for technical and medical expertise to set up secure data warehousing systems and manage multiple warehouse instances for various data-driven research projects. The platform simplifies the complex installation process and enables rapid instantiation of new instances of data warehousing systems, supporting both i2b2 and tranSMART. Additionally, this pipeline is designed to automatically detect the syntax and format of input data, handle different encodings, and manage missing and duplicate data, thus significantly reducing the efforts required for data cleansing and preprocessing. The solution was implemented and evaluated.

Vøgt et al. [54] proposed the development of an agile workflow to integrate climate data into the administrative process of urban land use planning, illustrated through the City of Constance as a case study. This process involves incorporating the Advanced Municipal Climate Data Store (AMCDS) toolbox to facilitate data-informed decision-making across various activities, including analysing the current situation and its effects, devising measures, testing and adapting them, and monitoring and assessing progress. The solution was developed and proposed, and the next steps involved presenting the workflow to administrative staff in participatory and co-creative workshops to gather feed-

back, which would then be integrated into an adjusted workflow. Sch"uttler et al. [51] proposed a solution to ensure the interoperability of the collected data by closely coordinating the definition and integration of datasets with MII. This process included translating the metadata of the sample, donor, and biobank datasets into HL7 FHIR profiles (Health Level Seven Fast Healthcare Interoperability Resources), standardising the relevant data within the established IT infrastructure and making them more accessible to query. Furthermore, the collaboration with BBMRI-ERIC resulted in significant synergies, enabling the quick technical integration of tools within the IT infrastructure, leading to international visibility and a high degree of interoperability. The solution was implemented and evaluated. The evaluation showed positive outcomes, confirming the system is user-friendly and technically feasible.

Semantic Heterogeneity: Semantic heterogeneity refers to the challenge of integrating data with divergent interpretations and meanings. Rosenkranz et al. [17] highlighted the problem of semantic heterogeneity during the design of data integration requirements in the development of Data Warehouses (DW). Dos Santos Júnior et al. [42] discussed the difficulty in accessing, integrating, analysing, and viewing data across heterogeneous applications with varying semantics. Abdallah and Fan [45] discussed semantic heterogeneity in heterogeneous MRO systems. Barcellos [27] discussed the difficulty of integrating data from different agents and tools due to semantic heterogeneity. Rix et al. [46] discussed the difficulty of integrating and making sense of diverse data sources with varying formats and meanings within the high-pressure die casting (HPDC) manufacturing process.

To resolve the semantic heterogeneity challenges, Rosenkranz et al. [17] proposed a communication-centric approach that combines agile software development practices with communication theory. This approach includes three core artefacts: a detailed template for data field specifications, a procedure model for iterative development, and the use of agile methods to facilitate continuous stakeholder communication. These components work together to enhance stakeholder comprehension and ensure consistent data usage across the organisation. The solution was implemented and evaluated. Dos Santos Júnior et al. [42] proposed developing an ontology-based approach known as *Immigrant* as a solution. By mapping the diverse semantics of various appli-

cations onto a common ontology, *Immigrant* seeks to resolve semantic conflicts and provide a cohesive and integrated view of the data. This enables the presentation of meaningful information through dashboards, facilitating data-driven decision-making in agile software organisations. The approach is currently in development, with plans for a proof of concept to validate its effectiveness. In addition to the data harmonisation and interoperability challenge addressed by Abdallah and Fan [45] suggested the development of ontologies discussed above to resolve data harmonisation, interoperability, and semantic heterogeneity in heterogeneous MRO systems.

Barcellos [27] proposed using ontologies to establish a common conceptualisation across different tools and data sources. This approach helps to reduce semantic conflicts and enables correct data integration by acting as an interlingua to map the concepts used by various agents and tools, thus facilitating understanding and communication. They developed and used domain ontologies as reference models to integrate tools that support Continuous Software Engineering (CSE) processes. They started by developing and integrating a Scrum Reference Ontology with ontologies from the Software Engineering Ontology Network (SEON), covering requirements and project management aspects. This ontology was then applied as a basis to integrate tools such as Clockify and Azure DevOps used by development teams in the software unit of a Brazilian government agency. As a result, data from different tools were integrated and displayed in dashboards, providing useful information for managers to make decisions. Rix et al. [46] proposed a solution that implemented a modular and agile information processing framework. This framework utilises annotation services, which add metadata to data sources, and a Master Data Management (MDM) repository, a centralised database that stores and manages master data. It enables real-time data streaming for data mining analyses and visualisation of results. The framework ensures that data from various sources are annotated with metadata, facilitating pervasive traceability and simplifying information aggregation. Additionally, using dynamic Object-Relational Mapping (ORM) frameworks, such as Hibernate, in conjunction with open interfaces like OPC-UA (Open Platform Communications Unified Architecture), addresses semantic heterogeneity. This is accomplished by enabling the labelling and exchange of information in a unified manner. The solution was implemented and evaluated at the Audi testing foundry.

Data Transformation and Extraction: Data transformation and extraction involve converting and transferring data from various sources into a usable format for analytics and decision-making. Hofer et al. [49] the challenge of extracting and refining data from semi-structured, mixed-quality crowd-sourced data such as Wikipedia and Wikidata. Dursun et al. [53] emphasised the need for a unified view of data from various sources and to integrate data from heterogeneous stores into a single, coherent data store for analytics and visualisation. Kannan et al. [48] discussed the importance of an efficient ETL process for extracting data from Electronic Health Records (EHR) into an Enterprise Data Warehouse (EDW). Dharmapal et al. [52] highlighted the complexity of integrating diverse data types without complex, time-consuming IT engineering efforts and suggested that modern analytic processing tools should transition away from traditional, retrospective BI tools and platforms towards more progressive analytic platforms.

To address the data transformation and extraction challenges, Hofer et al. [49] proposed a solution for improving the DBpedia Information Extraction Framework (DIEF) through the implementation of advanced ETL procedures. This approach included a systematic and test-driven method for handling data and code-related issues using Linked Data, which enhanced traceability and issue management. Additionally, the solution introduced two key technical enhancements: explicitly associating data artefacts with the corresponding code and implementing a library for continuous testing and validation of the data extraction process. The solution was implemented and evaluated. Dursun et al. [53] proposed implementing a data visualisation layer to simplify user access to data, thereby abstracting the complexities of different storage structures and technologies. In addition, it suggested the establishment of a data federation layer to integrate data from various autonomous stores into a single cohesive data store for users. Furthermore, the proposed solution included an extensive modelling process. This process encompasses defining the goal to comprehend the business value, preparing the data, selecting the type of data-driven analytics, choosing suitable tools such as data mining, machine learning, optimisation, and fuzzy inference systems, selecting tasks and algorithms from each tool, building the model, validating the model, and ultimately deploying the model. Kannan et al. [48] proposed developing a replicable ETL process

that allowed the extraction of data from the EHR into the EDW using standard EHR fields and a core set of EHR structures for capturing custom data. This process supported the parallel development of EHR-based registries and associated data collection tools. The solution was implemented and evaluated. Dharmapal et al. [52] suggested a shift from conventional, retrospective business intelligence (BI) tools towards more advanced analytic platforms that support seamless integration with diverse data sources, including external ones. It further recommended adopting an agile methodology to adapt flexibly to changes in data sources, methodologies, algorithms, or tools to achieve business goals.

Managing Data Integration: Vestues et al. [14] discussed the difficulty of managing data across various silos within the organisation and supporting analytical solutions that require cross-organisational insights. Little et al. [50] mentioned that a key business value of Landmark Graphics (a leading supplier of software and services) application suite arises from the integration of these products through a standard data model with over 800 tables, 12 entities, and 90 data types. This complexity presents a challenge in managing and integrating data across different products. Chen et al. [16] discussed the challenge of effectively managing internal and external data integration in big data projects, including data from existing EDW and new NoSQL systems.

To address the challenges related to managing data integration, Vestues et al. [14] proposed the adoption of a "data mesh" approach. This approach comprises four core principles: domain-oriented decentralised data ownership and architecture, data as a product, self-serve data infrastructure, and federated computational governance. The data mesh model advocates for a shift from centralised data management to a distributed model where domain teams have increased ownership of the data produced by their applications, aiming to improve coordination and enable more agile and automated approaches to data analytics. Vestues et al. [14] focused on reporting findings from a case study of a public sector organisation in Norway that has begun the transition from centralised to distributed data management. Little et al. [50] presented that Landmark Graphics adopted the Microsoft Solutions Framework (MSF), a milestone-based iterative development framework, to standardise their development process across different product teams. This framework underscores the importance of consistency and managing diverse data types effectively, consid-

ering the challenges of working with a unified data model. Chen et al. [16] adopted an architecture-centric approach, which strongly emphasises designing and developing a system's architecture as a crucial element of the development process. This approach entails the creation of a well-defined architecture, which acts as a blueprint for the system's development. It guides the selection of technologies, allocation of resources, and risk management throughout the development life cycle. This approach complements agile's modular design, a software development approach that emphasises breaking down a project into smaller, more manageable modules or components to manage the integration of diverse data types more efficiently. The approach includes using data lakes and cloud storage to handle various data types, supporting agile's requirement for flexible data structures and continuous delivery. The new methodology, AABA (Architecture-centric Agile Big Data Analytics), was implemented and validated through multiple case studies encompassing 10 big data analytics projects.

Diverse and Decentralised Data Sources: Upender [44] discussed the complexity involved in integrating patient data and transitioning to a centralised data management system within a diverse and decentralised user environment. Harper and Dagnino [41] highlighted the need to merge and clean data sets from different sources to enable advanced data analytics. Initiating and sustaining processes while ensuring data exchange and requests between components is challenging. The proposed solution is to distribute work requests across computing resources. Chhillar and Sharma [43] discussed the challenge of consolidating reports from various sources, such as development, testing, and bug-tracking tools, into a unified data analytics tool. This challenge pertains to seamlessly integrating diverse data sources and formats to generate real-time metrics and reports for software quality analysis.

To address the diverse and decentralised data sources challenges, Harper and Dagnino [41] proposed developing a standard product line architecture with built-in capabilities for data cleansing and integration, designed to serve a wide range of domains and reduce startup costs for new analytics applications. The solution was implemented as part of an agile evolutionary approach to build the advanced analytics product line architecture for various industrial application scenarios at ABB (ASEA Brown Boveri), a multinational corporation. Chhillar and Sharma [43] proposed an Automated Continuous

Table 4.3: Summary of Data Collection Challenges and Solutions

Data Collection Challenge	Data Collection Solution	Solution Status	Ref.
Capturing Diverse Data	User-centered design strategies for data collection	Implemented	[18]
	Continuous Integration servers and other software development tools	Proposed	[2]
	Analytics-Driven Testing (ADT) process	Proposed	[9]
Data Collection Method Challenges	Automated Continuous Testing TestBot and ACQ Metrics dashboard	Partially Implemented	[43]
	Automation of data collection and visualisation toolchain	Proposed	[58]
	Human-centred Agile Software Engineering (HASE) platform	Proposed	[20]
	Automated methods for data collection and visualisation	Proposed	[57]
	Centralised data management system with Scrum and XP practices	Proposed	[44]
Data Sharing and Collaboration	Diagnostic model for understanding data collection and sharing practices	Proposed	[8]
	Data-Driven Systems Engineering (DDSE) methodologies	Proposed	[21]
	Legal advisor for data sensitivity assessment	Proposed	[15]
Informative Data collection challenges	Development of ontology-based approach 'Immigrant'	Proposed	[42]
	Integration of development project data into Retrospective agendas	Proposed	[19]
	Q-Rapids tool for acquiring valuable information	Implemented	[56]
	Agile methodology and Data Analyst collaboration for data collection	Proposed	[52]
Comprehensive Data Collection	Creation of a generalised dataset and standardised data processing	Proposed	[60]
	Big data stack with scalable storage technologies	Proposed	[55]
	Project Management Information System for automated data collection	Proposed	[62]
	Automated data collection mechanisms from software in use	Proposed	[61]

Quality (ACQ) Metrics dashboard. This dashboard integrates data from various sources, such as functional, performance, security testing, Continuous Integration and Deployment (CI/CD) build details, and development reports into a data analytics tool. This integration facilitates the auto-generation of quality metrics, enabling real-time tracking and graphical representation of metrics. Chhillar and Sharma [43] also noted that further endeavours are necessary for its implementation. Upender [44] proposed a solution, discussed in Section 4.3.

4.3 Data Collection Challenges and Solutions

We have identified multiple challenges associated with data collection in agile development. These challenges and their proposed solutions are grouped into categories, which we discuss below. Furthermore, a summary of these challenges and their corresponding solutions is presented in Table 4.3.

Capturing Diverse Data: Pater et al. [18] focused on capturing data for personalised healthcare and human services while addressing barriers to widespread adoption and standardisation of information systems within healthcare. Matthies and Hesse [2] discussed the need for effective data collection and analysis from diverse sources, including software project data, customer feedback, and operational metrics. Furthermore, Batarseh and Gonzalez [9] highlighted the issue

of data is often not collected in a structured and organised manner, making it difficult to analyse and use for decision-making.

To address capturing diverse data challenges, Pater et al. [18] proposed a solution that involves leveraging a variety of user-centred design strategies, such as participatory design, empathic design, and design thinking, to guide the data collection methods. This approach maintains the user at the centre of the process and continuously incorporates user-driven input into each phase of the agile development cycle. The solution was implemented to modernise the Army Community Service's information technology infrastructure. Matt ies and Hesse [2] highlighted tools such as Continuous Integration servers, static analysis tools, and other software development tools that provide valuable data points on the current status and health of the developed software project. They highlighted that combining data analysis and interpretation by agile teams can enable better-informed business and software development decisions. Batarseh and Gonzalez [9] proposed the Analytics-Driven Testing (ADT) process. ADT involves a structured data collection and analysis approach within the agile software development lifecycle. During each sprint, data are systematically collected using the ADT tool. This includes recording all software failures, measuring the Mean Time Between Failures (MTBF), and collecting other relevant data such as software module, sprint number, and run time.

Data Collection Method Challenges: Chhillar and Sharma [43] discussed the difficulty in generating real-time software quality metrics and the need for actual data collection processes. In contrast, Lehtonen et al. [58] discussed the manual nature of data collection from the issue management system, which can be time-consuming and susceptible to errors. Upender [44] discussed the challenge involves creating a system that allows for the electronic capture of clinical research data, replacing the existing rudimentary tools such as Excel and Access that various independent research groups use. Lin et al. [20] discussed the difficulty of collecting data on software engineering activities, such as task allocation, collaboration, and mood stability, without disrupting the normal Agile Software Development (ASD) process. Practitioners surveyed by Svensson et al. [57] expressed concerns regarding data availability, excessive quantity, and ambiguity regarding usability, relevance, and its connection to decision-making.

To address the challenges related to the data collection method,

Chhillar and Sharma [43] proposed an Automated Continuous Testing (ACT) TestBot and ACQ Metrics dashboard. The ACT TestBot automates the execution of tests and the monitoring of application logs, while the ACQ Metrics dashboard aggregates and visualises the data, enabling the automatic generation of quality metrics reports. This integrated approach ensures the reliability of data and facilitates the real-time tracking of software quality. Chhillar and Sharma [43] noted the implementation requires more effort. Lehtonen et al. [58] recommended automating the data collection by implementing a toolchain that covers all steps from the initial data collection to visualisation. This automated approach would enhance the accuracy and efficiency of data collection, thereby improving the quality of the visualisations used to analyse the software development process. Additionally, the development of an interactive visualisation tool was suggested to recognise patterns and anomalies in the process, facilitating a deeper understanding and validation of the evolution towards a more agile process.

Lin et al. [20] presented a Human-centred Agile Software Engineering (HASE) platform. The HASE platform is an online Agile Project Management (APM) tool allowing unobtrusive data collection during the normal ASD process. It collects data on software engineering activities from the participants, such as task allocation, collaboration, and mood stability, without requiring additional effort from the participants. This approach ensures that the data collection process does not disrupt the regular workflow of the software engineering teams while still providing accurate and objective data for assessing software engineering skills. Svensson et al. [57] underscored that future research should develop novel automated methods for collecting, analysing, and visualising data to enhance existing agile decision-making processes by associating pertinent data with specific scenarios. Upender [44] proposed a three-year project to build a centralised data management system that addresses the needs of electronically collecting and integrating data with a gradual adoption and adaptation of Scrum and XP (Extreme Programming) practices. They emphasised a "just in time design" approach, which meant they approached the design phase with a flexible mindset. They used an evolutionary design process to handle ambiguous and conflicting requirements. Initially, they built a basic version of the clinical application to ensure it had essential functions, leaving the implementation of more advanced features for future updates. At the time of this research, they

were two years into the project.

Data Sharing and Collaboration: Fabijan et al. [8] noted the lack of customer and product data sharing among agile team members. This leads to difficulties in accessing data collected by others in different development phases and fragmented collection and storage of data. Kaur et al. [21] pinpointed concurrent access as a significant challenge, leading to unnecessary rework due to human error and inefficient communication methods. Barbala et al. [15] discussed the challenge of the uncertainty surrounding what data agile product teams are allowed to gather.

To address the data sharing and collection challenges, Fabijan et al. [8] developed a model that serves as a diagnostic tool to understand data collection and sharing practices within software development organisations. This model identifies the types of data collected, the parties responsible for collection, and the development phases in which the data is used. It also highlights critical handovers where data loss can occur. The model can provide valuable insights for companies deciding on actions to improve their data sharing and collection practices. This can aid in addressing the fragmented collection and storage of data, ultimately leading to better data analysis and utilization. Kaur et al. [21] recommended using Data-Driven Systems Engineering (DDSE) methodologies that facilitate concurrent access, enabling real-time collaboration and information exchange. DDSE tools manage engineering data during the implementation phase, provide version control, make it available collaboratively, and ensure full traceability for the entire engineering team. Barbala et al. [15] presented that including a legal advisor in the team helped assess the sensitivity of data when collecting, sharing, and storing it and provided guidance on compliance with data protection regulations.

Informative Data Collection: Dos Santos Júnior et al. [42] delved into difficulty in accessing, integrating, analysing, and viewing data handled by heterogeneous applications, which often adopt different semantics (the same information item is given divergent interpretations), posing a barrier to integrated data usage. Matthies [19] discussed the challenge of relying on subjective data gathered from team members' perceptions and experiences during Retrospective meetings. While readily available and relevant to team satisfaction, this subjec-

tive data may not provide a comprehensive or objective view of the team's performance and areas for improvement. Martínez-Fernández et al. [56] addressed the data collection challenge of obtaining informative user data. Dharmapal and Sikamani [52] underscored the significance of collecting data from suitable sources and filtering out unwanted data in the development phase.

To address the informative data collection challenges, Dos Santos Júnior et al. [42] proposed a solution "*Development of ontology-based approach Immigrant*" discussed in Section 4.2. This solution resolved the semantic heterogeneity data integration challenge and assisted in collecting informative data that can be used for data-driven decision-making in agile software development. Matthies [19] recommended integrating development project data and insights from software repository mining into Retrospective agendas. By employing project data from the last iteration in Retrospective meetings, teams can gain additional insights based on their team-specific data. This approach aims to provide a more thorough overview of the team's state, including facts and feelings, which can lead to better results in Retrospectives. Project data measurements can help track progress on common agile and software engineering challenges, enabling teams to identify and tackle improvement actions based on empirical project evidence rather than solely on anecdotal experiences.

Martínez-Fernández et al. [56] proposed using the Q-Rapids tool for acquiring valuable information for users. The Q-Rapids tool is designed to gather diverse data related to software system development and usage. These data are then structured within a Quality Model (QM) to analyse aggregated quality-related strategic indicators. These indicators are made available to decision-makers through a multi-dimensional and navigational dashboard employed during ASD events like sprint planning or daily stand-up meetings. The Q-Rapids solution was implemented and used in a real-world setting. Martínez-Fernández et al. [56] described a case study across four companies involving 26 practitioners to investigate the integration of a Quality Model within the Q-Rapids software analytics tool. Dharmapal and Sikamani [52] proposed utilising a combination of agile methodology principles and the expertise of a Data Analyst. During the planning stage, the Product Owner and the Data Analyst collaborate to select suitable Big Data Analytics tools and strategise the data source collection. This approach allows for incremental data collection in response to specific needs. In the development phase, the Data Analyst is pivotal in analysing data gathered

from various sources. They integrate data from diverse origins into a unified dataset, merge information based on shared attributes, and present it in a consolidated format. An essential aspect of their role is removing irrelevant data, employing an elimination technique that saves time and resources by excluding unnecessary data from processing.

Comprehensive Data Collection: Das et al. [60] delved into the challenges associated with collecting and storing substantial amounts of data for specific research purposes, mainly when the data is non-traditional, or the dataset is immense. Huang et al. [55] focused on efficiently ingesting and governing large volumes of heterogeneous data from various sources in various formats. Fagarasan et al. [62] stressed the importance of systematic and comprehensive data collection, beginning at the lowest level of project management. Olsson [61] discussed that traditional data collection methods, such as surveys and interviews, are insufficient for continuous experimentation.

To address the comprehensive data collection challenges, Das et al. [60] advocated for a two-fold approach: first, the creation of a generalised dataset aimed at expanding data coverage (not volume) by incorporating additional attributes. This approach ensures that a single data extraction step can serve multiple projects, with adjustments made in subsequent workflows to cater to specific research task requirements. Second, they emphasised standardised data processing, which involves abstracting standard data procedures, such as noise reduction and language standardisation, into reusable solutions and tools applicable across various data-driven research endeavours. This approach allows for systematic data collection and analysis, enabling more effective integration across various research projects. Huang et al. [55] The solution they proposed was to design a big data stack with scalable storage technologies and a flexible architecture. This stack included a data lake for raw data storage and integrated relational databases for real-time applications. Apache Spark was used as a distributed data computation engine to enable parallel processing for analytics tasks. The stack's design allowed for swift data analysis and agile development of data products, reducing the time required to answer analytical questions and develop applications. Fagarasan et al. [62] proposed using a market-available Project Management Information System (PMIS) to collect and analyse data systematically. This system enables the automated collec-

Table 4.4: Summary of Data Quality Challenges and Solutions

Data Quality Challenge	Data Quality Solution	Solution Status	Ref.
Ensuring Data Accuracy and Consistency Across Varied Sources	Solution referenced in Section 4.2	Implemented	[17]
	Solution referenced in Section 4.2	Implemented	[41]
	Solution referenced in Section 4.2	Implemented	[50]
	Solution referenced in Section 4.3	Implemented	[18]
	Solution referenced in Section 4.2	Implemented	[48]
Missing Quality Data	Utilising a quality-aware rapid software	Implemented	[6]
	Solution referenced in Section 4.3	Proposed	[58]
	Solution referenced in Section 4.3	Proposed	[56]
Inadequate Data Quality Management	Utilising earlier test-driven approaches	Proposed	[4]
	Solution referenced in Section 4.2	Implemented	[49]
	Solution referenced in Section 4.2	Implemented	[57]
Data Quality Standardisation	Solution referenced in Section 4.3	Implemented	[47]
	Solution referenced in Section 4.3	Proposed	[9]

tion and reporting of key performance indicators, crucial for monitoring project performance and ensuring cost-effective execution. The PMIS is a foundation for organising projects into a measurable portfolio, with performance metrics deeply embedded in the software development workflow. Olsson [61] proposed implementing mechanisms for automated data collection directly from the software in use, particularly in the post-deployment stage. This approach allows for the collection of real-time and actionable data that can be used for analysis to improve current products and inform the development of future ones, aligning with legal requirements for data utilisation and user consent.

4.4 Data Quality Challenges and Solutions

We have identified multiple challenges associated with data quality in agile development. These challenges and their proposed solutions are grouped into categories, which we discuss below. Furthermore, a summary of these challenges and their corresponding solutions is presented in Table 4.4.

Ensuring Data Accuracy and Consistency Across Varied Sources: Rosenkranz et al. [17] highlighted the importance of data quality (defined in their study as completeness, ambiguity, meaningfulness, and correctness) in application systems and project performance in IT or DW projects. The challenge lies in ensuring the accuracy and consistency of data from disparate sources. Harper and Dagnino [41] also emphasised the need to merge and clean datasets from different sources. The challenge here is ensuring the accuracy and con-

sistency of data for analysis. Little et al. [50] mentioned that the products are released regularly, with release cycles ranging from 3 to 18 months. This regularity poses a challenge to data management and to ensuring consistency across different product versions. Pater et al. [18] underscored the necessity to eradicate data redundancy and business process fragmentation in the context of IT system modernization. The challenge lies in aligning the diverse IT needs of users with the broader organisational data requirements. Kannan et al. [48] highlighted the challenge of ensuring the completeness, accuracy, and internal consistency of the critical few EHR fields essential for registry inclusion and eCQM (electronic Clinical Quality Measure) calculation.

To address the data accuracy and consistency challenges across varied sources: The solutions are the same as those of the data integration and collection (as those challenges co-occur together). We discussed those solutions in detail in Sections 4.2 and 4.3.

Missing Quality Data: Franch et al. [6] emphasised the issue when necessary quality data is unavailable (i.e., inaccessible, incomplete, inconsistent, and incorrect data) due to the absence of suitable data collection tools or processes or inaccessible data storage systems. For instance, SonarQube (mentioned as an existing tool that focuses on product quality or continuous integration) does not furnish raw data for specific base metrics. Lehtonen et al. [58] highlighted potential data quality problems in software engineering data, including noise, outliers, low precision, missing values, coverage errors, and clones. An example given is the potential for inaccurate data due to developers forgetting to update the issue management system task state when development work starts or ends, leading to inaccurate timestamps. Martínez-Fernández et al. [56] discussed the need for transparency and clarity on the raw data used to compute factors and indicators in the software analytics tool (Q-Rapids tool). End-users like product owners may require a clear understanding of the raw data used to derive specific values and the associated decision-making processes.

To address the missing data Franch et al. [6] discussed the Q-Rapids project, mentioned as an initiative aimed at developing a quality-aware rapid software development methodology. The Q-Rapids project addressed data quality challenges by selecting and configuring tailored data collection tools aligned with business objectives, specifying precision requirements, implementing real-

time checks to prevent errors, and integrating diverse data sources while transforming them for compatibility with specific analysis methods and tools. The solution was implemented, evaluated, and refined through three releases deployed by the four industry partners in the project. Note that the solutions suggested to resolve the remaining data quality challenges, as discussed by Lehtonen et al. [58] and Martínez-Fernández et al. [56], are the same as those of data collection (as they co-occur together). We discussed those solutions in detail in Section 4.3.

Inadequate Data Quality Management: Ambler [4] highlighted that data quality issues cost US organisations an estimated 600 billion dollars annually (according to the Data Warehouse Institute), stemming from the use of traditional data management approaches to ensure data quality within organisations, according to the Data Warehouse Institute as indicated by a survey conducted by Ambler [4]. Svensson et al. [57] emphasised that the quality of the data and the processing techniques and tools used to handle it will impact the quality of decisions made using Data-Driven Decision Making (DDDM). Hofer et al. [49] discussed the data quality challenge of ensuring the "fitness for use" of data, which is often neglected or delayed in the software engineering process until the end-user evaluation phase, known as the "*point-of-truth*". This delay impacts the cost-effectiveness of data quality management and contradicts agile principles that emphasise early and continuous delivery of valuable software.

To address inadequate data quality management Ambler [4] suggested that utilising agile methods, which typically involve a more and earlier test-driven approach to development, results in better quality with more and early testing. Additionally, developers and data professionals should apply concrete, quality-focused techniques for evolutionary development. Note that the solutions suggested to resolve the remaining data quality challenges, as discussed by Svensson et al. [57] and Hofer et al. [49], are the same as those of data collection and data integration (as they co-occur together). We discussed those solutions in detail in the data collection Section 4.3 and for data integration Section 4.2.

Data Quality Standardisation: Spengler et al. [47] mentioned the need for significant data restructuring and cleansing due to heterogeneous data to ensure suitability for proper integration into clinical data warehousing solutions.

Table 4.5: Summary of Data Analysis Challenges and Solutions

Data Analysis Challenge	Data Analysis Solutions	Solution Status	Ref.
Analysing Large and Complex Data	Solution referenced in Section 4.2	Proposed	[16]
	Solution referenced in Section 4.3	Proposed	[52]
	Solution referenced in Section 4.3	Proposed	[60]
	Solution referenced in Section 4.3	Proposed	[20]
	Well-documented tools and developer training.	Proposed	[63]
Analysing Semantic Heterogeneity Data	Solution referenced in Section 4.2	Proposed	[42]
	Solution referenced in Section 4.2	Implemented	[27]
	Solution referenced in Section 4.2	Implemented	[46]
Efficient Data Analysis and Visualisation	Solution referenced in Section 4.2	Proposed	[53]
	Solution referenced in Section 4.3	Proposed	[57]
	Solution referenced in Section 4.3	Proposed	[58]
	Solution referenced in Section 4.3	Proposed	[8]
Real-Time Data Analytics and Decision Making	Solution referenced in Section 4.2, Section 4.3	Proposed	[43]
	Solution referenced in Section 4.3	Proposed	[61]
Selection of Appropriate Analytical Techniques	Solution referenced in Section 4.2	Implemented	[41]
	Solution referenced in Section 4.3	Proposed	[2]
	Solution referenced in Section 4.3	Proposed	[19]
	Solution referenced in Section 4.3	Proposed	[62]
	Solution referenced in Section 4.3	Proposed	[9]

Batarseh and Gonzalez [9] highlighted the need to enhance dataset quality, standardise variables, and eliminate unnecessary data. The challenge here involves dealing with poor-quality data and standardising data for analysis.

To address the data quality standardisation challenges: The solutions are the same as those of data integration and collection (as they co-occur together). We discussed those solutions in detail for data integration in Section 4.2 and for data collection in Section 4.3.

4.5 Data Analysis Challenges and Solutions

We have identified multiple challenges associated with data analysis in agile development. These challenges and their proposed solutions are grouped into categories, which we discuss below. Furthermore, a summary of these challenges and their corresponding solutions is presented in Table 4.5.

Analysing Large and Complex Data: Chen et al. [16] delved into the challenge of developing an extensive data system that effectively supports advanced analytics by addressing the 5Vs of big data (Volume, Velocity, Variety, Veracity, and Value) and facilitating effective collaboration between data scien-

tists and software engineers to maximize the value from big data. Dharmapal and Sikamani [52] underscored the challenge of managing and processing the increased volume of unstructured data, which requires more complex enhancements in data management and analysis techniques. Das et al. [60] discussed the challenge of analysing extensive volumes of data, particularly when the data is non-traditional or when the dataset is massive. Traditional analysis techniques often fall short due to these challenges. Lin et al. [20] discussed the challenge of the high dimensionality of the datasets, which included detailed interactions and decisions made by the participants (students in the study). This complexity made it difficult to identify which features (which could include metrics such as the number of collaborators per task, the frequency of task updates, mood stability, and other behavioural indicators) or a combination of features could accurately predict certain behaviours of interest. Hamer et al. [63] highlighted that interpreting the data provided by data-driven tools such as Git, Jira, and SonarQube can pose a significant barrier to beginning the effective use of these tools due to the large volume and complexity of the data generated. Furthermore, analysing this data necessitates training to utilise the information correctly.

To address the analysing large and complex data challenges: The solutions are the same as those of the data integration and collection (as they co-occur together). We discussed those solutions in detail for data integration in Section 4.2 and for data collection in Section 4.3. Additionally, Lin et al. [20] solution is to employ the application of Exploratory Data Analysis (EDA) to analyse the collected data. EDA is often used to summarise the main characteristics of data sets using visual methods. It aims to understand what can be learnt from the data beyond the formal modelling or hypothesis-testing tasks. Furthermore, Hamer et al. [63] solution recommends that these tools be well-documented and provide tutorials to facilitate easier adoption and usage. Developers should be trained to comprehend and leverage the provided information effectively. Future work could involve creating simplified visualisations that enhance information understanding. This proposed solution has not been implemented.

Analysing Semantic Heterogeneity Data: Dos Santos J'unior et al. [42] highlighted the difficulties organisations encounter when accessing, integrating,

analysing, and viewing data managed by disparate applications. Barcellos [27] discussed the need for continuous software measurement, which involves collecting and analysing data to provide useful information for daily activities and decision-making in software development. Rix et al. [46] addressed the challenge of analysing real-time data streams using data mining methods.

To address the analysing semantic heterogeneity data challenges: The solutions are the same as those of the data integration (as they co-occur together). We discussed those solutions in detail for data integration in Section 4.2.

Efficient Data Analysis and Visualisation: Dursun et al. [53] discussed the challenges of efficiently analysing and visualising vast and diverse oil and gas data and developing intelligent data-driven analytics software to optimise decision-making and minimize. Svensson et al. [57] addressed data analysis as a crucial step in data-driven decision-making (DDDM). The challenge lies in the fact that the quality of decisions is directly proportional to the quality of processing techniques and tools. In other words, ineffective or incorrect data analysis can lead to suboptimal or erroneous decisions. Lehtonen et al. [58] explored the utilisation of visualisations for data analysis. The challenge lies in the potential complexity of these visualisations, which may hinder understanding. Fabijan et al. [8] highlighted that companies are not capitalising fully on the data they amass. This is attributed to challenges associated with efficiently and meaningfully integrating and analysing customer data.

To address the efficient data analysis and visualisation challenges: The solutions are the same as those of the data integration and collection (as they co-occur together). We discussed those solutions in detail for data integration in Section 4.2 and for data collection in Section 4.3.

Real-Time Data Analytics and Decision Making: Chhillar and Sharma [43] discussed using data analytics tools for generating metric reports. These reports are vital for indicating the overall health of the build post-deployment. The challenge here is the requirement for real-time data analytics to support decision-making processes. Olsson [61] discussed the difficulty in handling and analysing the post-deployment data collected from embedded systems.

To address the real-time data analytics and decision-making chal-

lenges: The solutions are the same as those of the data integration and collection (as they co-occur together). We discussed those solutions in detail for data integration in Section 4.2 and for data collection in Section 4.3.

Selection of Appropriate Analytical Techniques: Harper and Dagnino [41] discussed the need to specify analysis strategy, process substantial amounts of data, and complete analyses promptly. The challenge lies in fulfilling these expectations while managing the complexity inherent in advanced analytics applications. Matthies and Hesse [2] discussed data generation during software development processes, such as work item descriptions, documentation, and version control information. The challenge lies in efficiently managing and analysing this data. Matthies [19] discussed using various metrics designed for agile practices or tools that developers are already familiar with, such as the git command line, for project data analysis. The challenge is that improvement actions are often based on anecdotal evidence and experiences rather than empirical project evidence. Fagarasan et al. [62] discussed the necessity for effective data analysis to extract meaningful insights from collected data. The challenge lies in interpreting, analysing, and converting this information into actionable insights. Batarseh and Gonzalez [9] underscored the need to define variables for model development and investigate different models. The challenge here lies in selecting appropriate variables and models for analysis.

To address the selection of appropriate analytical techniques challenges: The solutions are the same as those of the data integration and collection (as they co-occur together). We discussed those solutions in detail in Sections 4.2 (data integration) and 4.3 (data collection).

4.6 Other Data Management Aspects Challenges and Solutions

Several challenges and solutions were identified across the other 11 data management aspects. We briefly discuss those challenges and solutions below. We provide a summary in Table 4.1. Each of these aspects presents challenges that require specific solutions, which we will explore below.

- *Data Storage:* Challenges include the integration of data storage pro-

cesses and handling large data centres. Solutions involve technologies like MySQL, RRDtool, and Elasticsearch.

- *Data Validation and Governance:* Challenges include big data veracity and privacy issues. Solutions involve architectural design considerations and developing abstraction mechanisms.

- *Data Ingestion:* Challenges include real-time processing of raw data and integrated data usage from heterogeneous applications. Solutions like the Lambda architecture are proposed.

- *Data Visualisation:* Challenges include diversifying visualisation techniques and understanding the technologies involved. Solutions involve developing a common architecture for data visualisation.

- *Data-Driven Decision Making (DDDM)*: Challenges include transitioning from non-data-based decision-making processes and scaling managerial capabilities. Solutions involve organising ontologies in a network and improving data analysis.

- *Data Privacy:* Challenges include ensuring compliance with regulations like GDPR and addressing ethical issues in data collection. Solutions include using data models for privacy requirements and aligning data collection with legal requirements.

- *Data Security and Compliance:* Challenges include providing secure access to data and maintaining audit trails. Solutions involve tools like the Rational Unified Process and encryption technology.

- *Data Testing:* Challenges include establishing a known state for the database and managing analytics-driven testing. Solutions include using tools like dbUnit and continuous data analysis.

- *Data Management Development:* Challenges include agility in data model design and software effort in system development. Solutions include adopting agile approaches and using tools like MagicDraw.

- *Data-Driven Development:* Challenges include assimilating data science roles into agile teams and utilising data in effort estimation. Solutions involve interdisciplinary competence and advanced intelligent approaches.

- *Data Maintenance Strategy:* Challenges include responsiveness to data requests (the ability to quickly provide accurate and relevant data when needed) and creating data products (producing data outputs or reports that support decision-making and meet user needs). Solutions involve improving agility in data management practices and coordination of people, processes, and technology.

4.7 Types of Data Management Challenges

Data management challenges in agile software development can vary widely, often influenced by the specific needs of agile practices, the domain in which they are applied, or general data management principles. We classified these challenges into three distinct types: Agile-Intrinsic Challenges, Domain-Specific Challenges, and General Data Management Challenges.

- *Agile-Intrinsic Challenges* refer to issues that arise specifically from the principles and practices of agile methodologies, such as the need for frequent updates or rapid feedback cycles.
- *Domain-Specific Challenges* are unique to specific industries or application areas and may not originate from agile practices but rather from the data needs and complexities inherent to those fields (e.g., climate data for government or maintenance data in aviation).
- *General Data Management Challenges* are issues commonly occurring across various development methodologies and industries, such as ensuring data quality, consistency, and standardisation, regardless of whether agile practices are used.

Table 4.6 provides an overview of the data management challenges identified in this study, categorised into Agile-Intrinsic, Domain-Specific, and General challenges

Table 4.6: Types of Data Management Challenges: Agile-Intrinsic, Domain-Specific, and General

Challenges	**Agile-Intrinsic**	**Domain-Specific**	**General**	**Ref.**
Data Integration Challenges				
Data Harmonisation and Interoperability		X		[45]
Semantic Heterogeneity		X		[17]
Data Transformation and Extraction	X			[49]
Managing Data Integration			X	[14]
Diverse and Decentralised Data Sources		X		[41]
Data Collection Challenges				
Capturing Diverse Data		X		[18]
Data Collection Method	X			[43]
Data Sharing and Collaboration			X	[8]
Informative Data Collection	X			[42]
Comprehensive Data Collection			X	[60]
Data Quality Challenges				
Ensuring Data Accuracy and Consistency			X	[17]
Missing Data Quality		X		[6]
Inadequate Data Quality Management	X			[4]
Data Quality Standardisation			X	[47]
Data Analysis Challenges				
Analyzing Large and Complex Data			X	[16]
Analyzing Semantic Heterogeneity Data		X		[27]
Efficient Data Analysis and Visualization	X			[53]
Real-Time Data Analytics and Decision-Making	X			[43]
Selection of Appropriate Analytical Techniques			X	[62]

4.8 Summary

In this chapter, we present the results of the SLR, which identified data management challenges and solutions in agile software development across different data management aspects. Those include challenges and solutions across data integration, quality, collection, and analysis. Figure 4.3 summarises the challenges and links to solutions. To address the challenges identified in each aspect, we found that the proposed or implemented solutions vary across studies, depending on the nature of the systems and the maturity of the processes followed. We also note that most solutions are mainly proposed with no empirical evaluation.

To further explore these challenges and the solutions proposed in the literature, we conducted a survey with practitioners (presented next in Chapter 5). This survey aims to capture the practical insights from agile software development practitioners in various industries and roles, complementing the findings from the SLR.

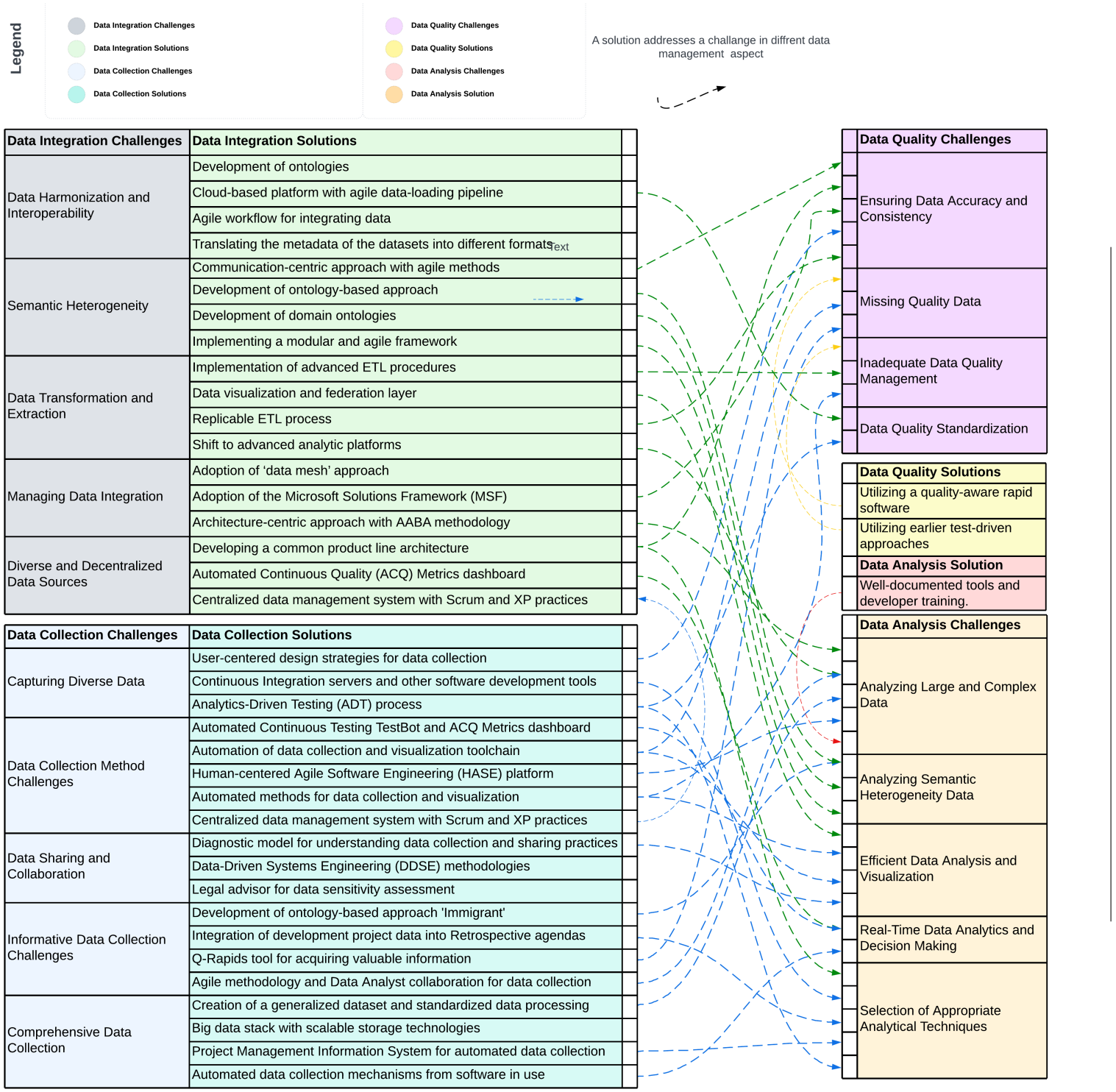

Figure 4.3: Summary: Data Management Challenges and Solutions from the SLR

Chapter 5

Data Management Challenges and Solutions Survey

In this chapter, we are seeking to gain practical perspectives from agile professionals in different roles, in line with the research questions posed at the outset of this thesis (RQ1: What are the data management challenges in agile software development? and RQ2: What are the proposed solutions to address these challenges?). The survey was distributed in March 2024, and the response period was open until June 2024. It was distributed to agile software development practitioners through networks, professional groups, and online platforms. A total of 32 responses were received, with the survey taking approximately 15-20 minutes to complete. The chapter first presents the demographic information of the survey participants, followed by a detailed analysis of their responses. The results are divided into four main sections based on the top four data management aspects we identified in Section 4.1 (data management: integration, quality, collection, and analysis). The chapter explores the challenges reported by practitioners and the solutions they have implemented or found effective for each aspect. A summary of the results of the practitioner survey is provided in the last section.

5.1 Survey Demographics

Figure 5.1 provides an overview of the survey participants, detailing key demographic information such as gender distribution, geographic location, and levels of industry experience. Below, we present the highlights of the demographics:

1. Age: Most participants are aged 35-44 (56.25%).

2. Gender: 28 participants (87.5%) are identified as male, and the remaining 12.5% as female.

3. Experience: 47% of the participants have more than ten years of experience, followed by 9 participants (28%) with 6-10 years of experience. This indicates that the majority of participants possess significant experience working in the industry, which can lend valuable insights to the challenges and solutions discussed in this thesis.

4. Education Level: Most participants have completed a bachelor's degree in a computing-related discipline (47%). Others have a Master's degree or higher (38%), a doctorate (9%), a degree not related to a computing-related discipline (3%), or indicated N/A (3%).

5. Agile Roles: the roles of the participants are varied, including developers (41%), product owners, senior project managers, scrum masters, and analysts.

6. Location: Practitioners came from eight countries (New Zealand, Saudi Arabia, UAE, France, Egypt, China, USA, and Germany), with most practitioners based in New Zealand (50%).

7. Industry Sectors: Most participants work in the ICT sector (44%). Other sectors (56%) include government administration-defence-public safety (31%), education (9%), manufacturing (6%), public transport (3%), healthcare (3%), and retail (3%).

8. Seniority Levels: Most participants hold senior-level positions (62%), followed by mid-level (19%) and executive positions (12%). Entry-level positions are the least represented (6%). That means most of the survey participants are seniors or higher (75%).

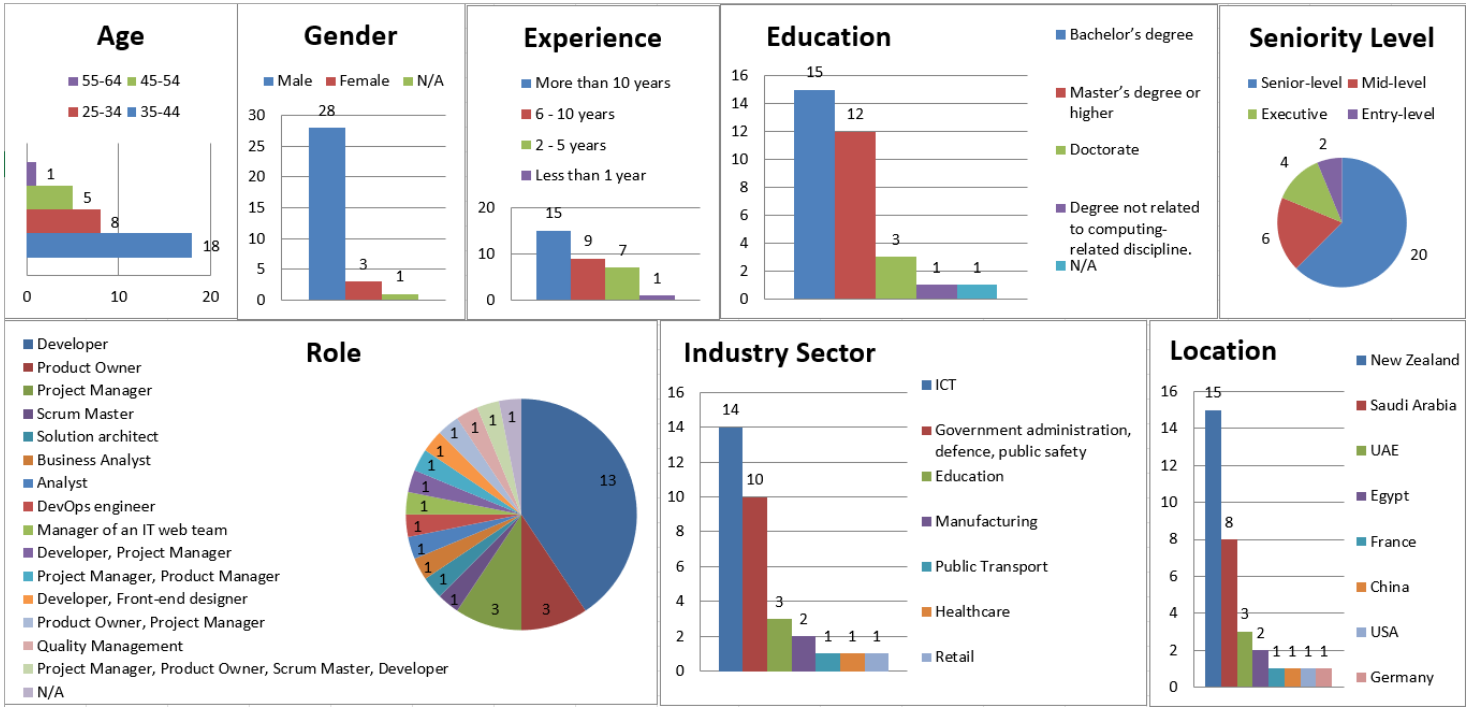

Figure 5.1: Survey Demographics

5.2 Data Management Challenges and Solutions

In the upcoming sections, we will explore the specific challenges associated with each aspect and present the solutions suggested in the survey to tackle them. We provide a detailed description of data management challenges and solutions across all four aspects in Appendices A and B.

5.2.1 Data Integration

The participants have flagged several challenges associated with data integration in agile software development. Figure 5.2 illustrates these challenges along with the number of participants who reported each one. Figure 5.3 illustrates the solutions and the number of participants who reported each one.

Challenges

The participants have flagged that *managing data integration processes* is the most common challenge they face (62.50%). For instance, managing data integration can be difficult when bugs and errors occur in the integration system, like when automated systems fail to sync data between tools, causing delays and inaccuracies. *Complexity of Integrating Real-Time Data* (44%) is the least frequently mentioned challenge, such as when integrating real-time user data into a dashboard requires constant updates and adjustments to ensure accuracy. Participants highlighted that the data integration process is challenging because of bugs and errors in the integration system used and possible data quality issues (such as data accuracy), which is crucial because inaccurate data can lead to faulty analysis, incorrect decision-making, and reduced trust in the system. Additionally, data security issues (e.g., inaccessible data due to privacy concerns) and unclear integration requirements at the beginning of the project, which keep changing, further complicate the process.

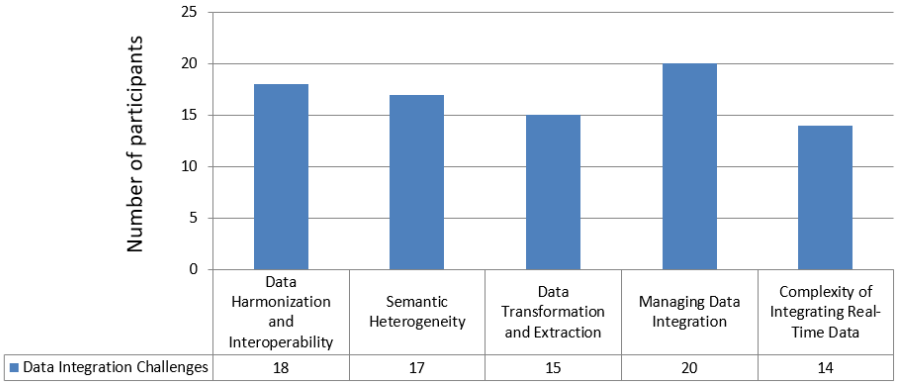

Figure 5.2: Data Integration Challenges

Solutions

Participants have identified the use of *communication centric approaches* as the most frequently adopted solution for data integration challenges (56%). *Automated continuous testing* is the least frequently mentioned solution (34%). Participants also employed solutions such as *customized technical solutions*. For example, one participant noted that "*custom scripts were developed to automate the transformation of data into the required formats and structures before loading them into the target system*". Another solution that practitioners noted is *standardisation and governance* of data. Additionally, *data segmentation and simplification* (i.e., dividing complex data integration into smaller ones) was also mentioned as a solution.

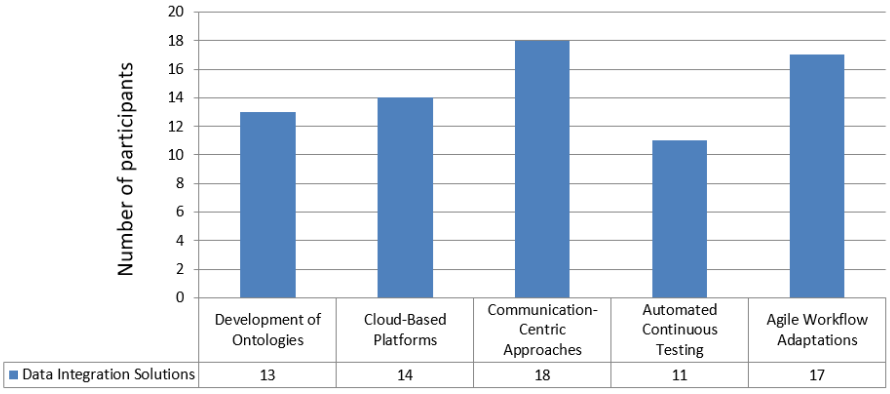

Figure 5.3: Data Integration Solutions

5.2.2 Data Collection

In agile software development, the process of collecting data comes with its own set of challenges, as reported by the participants. Figure 5.4 provides these challenges, along with the number of participants who identified each one. Additionally, Figure 5.5 captures the solutions that were suggested to address these data collection challenges.

Challenges

Participants have flagged that *capturing diverse data* (59%) and *automation challenges* (59%) are the most common data collection challenges. For example, capturing diverse data can be difficult when dealing with data from different sources, like social media and purchase histories, which use different formats. Automation challenges occur when automated systems, such as error logging tools, fail to correctly categorise data, leading to confusion in prioritising tasks. On the other hand, *comprehensive data collection* is the least frequently mentioned challenge (34%), where agile development teams might miss important user experience data while focusing on technical metrics. It was also stressed that the data collection is challenging because of the *quality of the data being collected*, which in some cases might be incomplete or irrelevant to the project. Other data collection challenges noted by participants include *time constraints* as the agile sprint time is insufficient to complete the tasks, and *privacy issues* such as unavailable data to be collected due to ethical considerations.

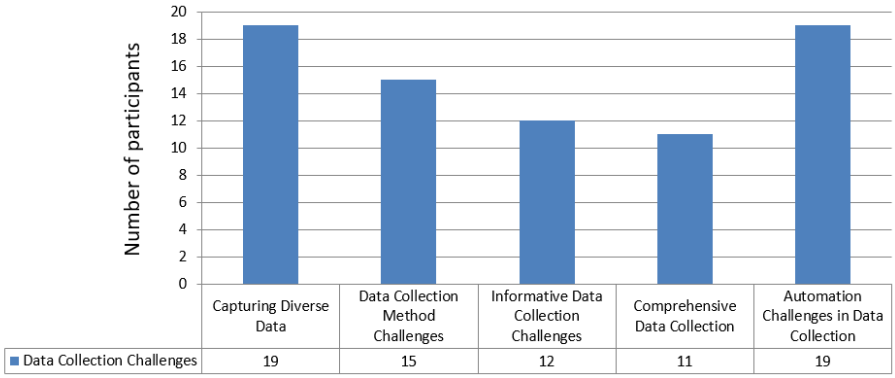

Figure 5.4: Data Collection Challenges

Solutions

Several data collection solutions have been identified in the survey, including the use of *user-centred design strategies* (59%) and *Automation of data collection and visualisation toolchain* (53%).*Q-Rapids tool for valuable information* (3%) is the least adopted solution. Participants highlighted the use of data transformation and standardisation as a solution. It was noted that transforming source data into a standardised format ensures compatibility between source and target systems.

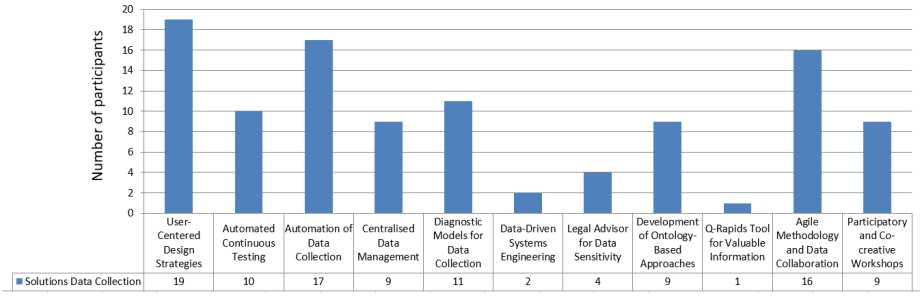

Figure 5.5: Data Collection Solutions

5.2.3 Data Quality

Data quality is a significant concern in agile software development, as highlighted by the participants. Various challenges were identified, as depicted in Figure 5.6, which shows the frequency with which each challenge was reported. Correspondingly, Figure 5.7 presents the strategies participants have adopted to address these challenges.

Challenges

Participants have indicated that *ensuring data accuracy and consistency* (66%) and *completeness of data* (66%) are the most common data quality challenges. For example, ensuring data accuracy and consistency can be challenging when different sources report inconsistent data, like when sales figures from one system differ from inventory records in another. Completeness of data is an issue when key information is missing, such as during a sprint review when user stories lack necessary acceptance criteria. On the other hand, *Effective data quality*

management (44%) is the least frequently mentioned challenge, which involves maintaining data integrity over time, like when automated systems struggle to correct data issues, requiring manual intervention.

Figure 5.6: Data Quality Challenges

Solutions

The participants have identified the use of *automated data cleaning and enrichment tools* (56%) and *standardisation of data quality metrics and processes* (56%) as the most frequently adopted solutions for data quality challenges. *Integration of advanced analytic platforms* is the least reported solution (9%). The participants also employed solutions such as *training and awareness*, which involves providing training on data quality best practices, and *data restructuring*, which involves modifying the structure of data to improve its quality.

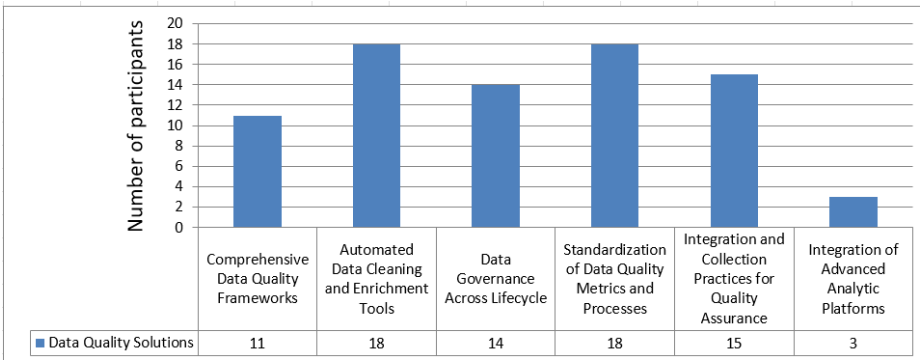

Figure 5.7: Data Quality Solutions

5.2.4 Data Analysis

Participants in the survey pointed out numerous obstacles related to data analysis within agile environments. The challenges they encountered are visualised in Figure 5.9, alongside the number of participants who mentioned each. Solutions to these challenges, as proposed by the participants, are illustrated in Figure 5.9.

Challenges

The participants have flagged *that complex data sets* (59%) and *real-time analysis requirements* (59%) are the most frequently reported challenges. For instance, dealing with complex data sets can be challenging when teams have to analyse large volumes of user data across multiple sprints, requiring constant refinement and processing. *Real-time analysis requirements* pose difficulties when teams need to analyse data as it is generated, such as when developing a real-time user engagement tracker that must continuously adapt to changing data. *ensuring analytical accuracy* (47%) is also a significant challenge, where teams might face issues maintaining precision in their analysis, like when pair programming is used to enhance accuracy but adds extra steps to the agile process. On the other hand, *Analyzing semantic heterogeneity* (the challenge of dealing with data that has different meanings across sources) (25%) and *selection of appropriate analytical techniques* (where teams must choose the best methods for analysis, are less frequently reported challenges) (25%) are the least reported challenges. The participants highlighted that data analysis is challenging because of *data integration and quality challenges*; for example, incomplete historical data due to integration difficulties or missing data can hinder the development of accurate predictive models. The participants mentioned that the *data collection and preparation issues* may also impact data analysis. For example, it was mentioned that there is consistently a need "*to gather various data from various resources to model them together to get meaningful insights*". Another data analysis challenge noted is *unclear objective and scope issues* such as situations where the goals and boundaries of the data analysis tasks are not well-defined.

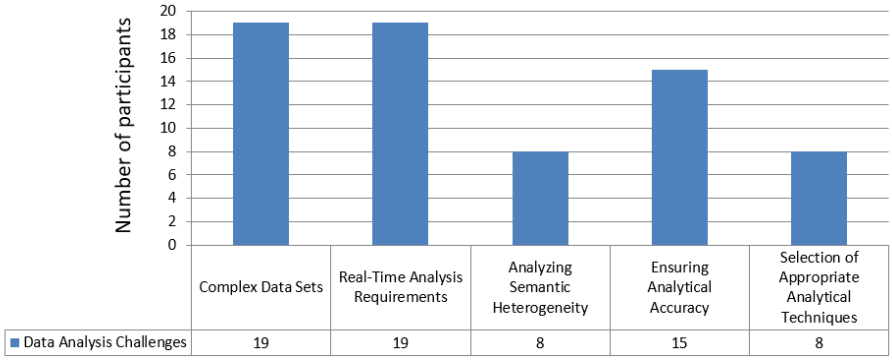

Figure 5.8: Data Analysis Challenges

Solutions

Compared to the findings from our SLR, the participants have identified the use of *advanced analytical tools and techniques* (50%) as the most frequently adopted solution for data analysis challenges. *Adoption of machine learning and AI for data analysis* (12.50%) is the least reported solution. The participants also highlighted that *improve data integration, data quality, and collection methods* can facilitate the data analysis process.

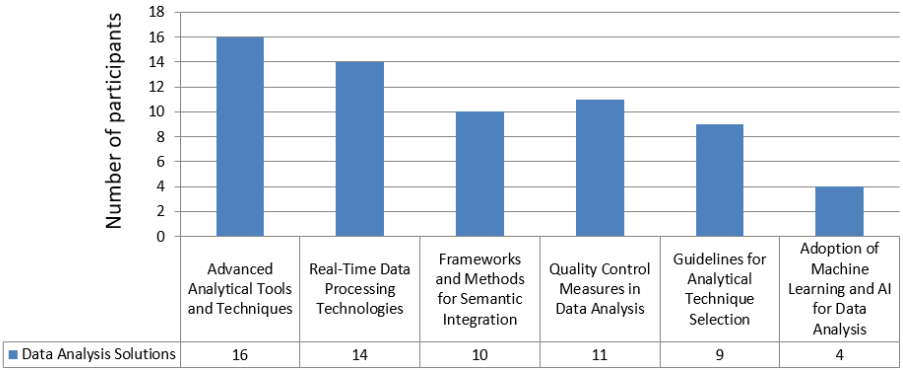

Figure 5.9: Data Analysis Solutions

5.3 Types of Data Management Challenges

The survey findings complement the challenges identified in the SLR, thereby reinforcing the classification of data management challenges presented in Chap-

ter 4.7. The survey results confirm the presence of similar categories—Agile-Intrinsic, Domain-Specific, and General Data Management Challenges—and expand upon them with additional insights.

For instance, the data integration challenge of *Complexity of Integrating Real-Time Data* can be classified as agile-intrinsic because agile projects often require real-time updates to support rapid feedback cycles, making real-time integration particularly demanding. Additionally, the data collection challenge of *Automation Challenges* in data collection also falls under agile-intrinsic as frequent iterations in agile benefit from automation to streamline workflows and reduce disruptions.

On the other hand, some challenges identified in the survey confirm classifications as general data management challenges. For example, the data quality challenge of *Completeness of Data* can be considered a general challenge, as ensuring all necessary data is present is a universal data quality concern, not exclusive to agile development.

5.4 Summary

Our survey was conducted to complement the SLR findings with insights from industry professionals. The survey involved 32 practitioners, predominantly senior-level professionals with over ten years of experience in agile software development. The survey confirmed many of the challenges identified in the SLR regarding data management aspects (RQ1). The results show that managing data integration processes, capturing diverse data, automation data collection challenges, ensuring data accuracy and consistency, completeness of data, complex data sets, and real-time analysis requirements are the most notable challenges faced by the participants. The participants adopted various solutions to address different data management challenges, which are mainly in line with the solutions reported in the SLR (RQ2). However, the survey results highlighted the need for training programs focusing on data management skills, which was not emphasised much in the previous studies. The survey findings show a greater need for real-time analytics and automation than indicated in the SLR. Detailed descriptions and examples of each challenge and solution are provided in Appendices A and B.

Chapter 6

Discussion

In this chapter, we discuss the findings from the SLR and the insights from the practitioner survey to provide a thorough analysis of data management challenges and solutions in agile software development. This chapter is divided into three main sections that provide comprehensive insights into the research questions posed at the outset of this thesis (RQ1: What are the data management challenges in agile software development? and RQ2: What are the proposed solutions to address these challenges?). The first section compares the findings from both the SLR and the practitioner survey. The second section discusses potential implications for agile process activities and project roles, identifies common challenges and workable solutions, and offers practical advice. The third section provides recommendations for practitioners to address and overcome data management challenges. Finally, the last section summarises the discussion presented in this chapter.

6.1 Comparison of the SLR and the Survey Findings

The survey results also reflect data management challenges identified in the SLR. Most participants reported challenges with managing data integration processes, emphasising the need to manage diverse data sources effectively. This aligns with the SLR's findings on the complexities of managing data integration across various systems. Similarly, most survey respondents highlighted

challenges with ensuring data accuracy, consistency, and completeness, which supports the SLR's emphasis on the importance of data quality for effective decision-making.

The survey offered additional practical insights into data collection and analysis challenges. Most practitioners reported challenges regarding capturing diverse types of data (such as structured, unstructured, and semi-structured data) during data collection and highlighted difficulties with automating data collection processes. These findings align with the SLR's emphasis on the need for comprehensive and standardised data collection methods. Regarding data analysis, most participants reported difficulties analysing complex datasets (which originated from the projects participants were actively working on) and the need for real-time analysis. This underlines the importance of advanced analytical tools and techniques, including communication-centric approaches and agile workflow adaptations, as solutions for complex data analysis.

The survey also showed a notable need for training programs that enhance data management skills within agile teams. This aspect was not emphasised much in the previous studies we analysed in the SLR. Practitioners highlighted challenges in implementing real-time data analytics, stressing the need for better tools and methodologies. The survey findings highlight a greater need for automation and real-time analytics than those mentioned in the SLR.

6.2 Implications

We explore the potential implications of our SLR and survey results. We discuss how these findings impact different agile process activities, including the impacted agile project roles and the potential strategies these roles can follow to mitigate possible challenges.

Requirements Gathering: Inaccurate or incomplete data during requirements gathering can lead to poorly defined project goals and user stories, negatively impacting the entire development process and potentially leading to unmet stakeholder expectations. Product owners may struggle to define clear project goals. To mitigate this, they should ensure effective stakeholder communication and use comprehensive data collection methods. Business analysts may struggle to gather and document project requirements accurately. Product owners should also employ structured data-gathering techniques, use appropriate tools, and

validate data with multiple sources. Ineffective data management, particularly in data collection, can result in poorly defined requirements, leading to misunderstandings and potential rework for the project team. To mitigate this, the team should collaborate closely with all stakeholders, thoroughly review requirements before implementation, and provide feedback to improve data collection methods.

Sprint Planning: Comprehensive data collection is crucial during sprint planning for defining sprint goals and prioritising tasks. Data management challenges (e.g., inaccurate data collection) can lead to poorly defined sprint goals, affecting prioritisation and sprint success and increasing the need for spike sprints (short, focused iterations designed to research or investigate a particular problem or uncertainty) [68]. Scrum masters may struggle to effectively facilitate the sprint planning session due to data inaccuracies. They should implement robust data validation processes, ensure clear communication of data management challenges, and facilitate collaborative problem-solving sessions. The product owners may face difficulties setting realistic sprint goals and prioritising tasks. They should use data analytics tools to improve data accuracy, prioritise high-quality data sources, and regularly review data collection methods. The agile development team may end up working on poorly defined or prioritised tasks. Therefore, they may need extended workshops to understand and address the data management challenges, consuming valuable time and resources. To mitigate this, they should communicate continuously with the scrum master and product owner, participate in data validation, and ensure a clear understanding of sprint goals.

Daily Stand-ups: Daily stand-ups are focused on identifying immediate obstacles and coordinating daily tasks. Data management challenges, such as incomplete data (e.g., missing key information during a sprint review, like user stories lacking necessary acceptance criteria), can impede the ability to quickly identify blockers and make informed decisions about the day's work. The project team may struggle to identify and communicate daily progress and blockers. They should use real-time data tracking tools, ensure daily stories data updates, and encourage open communication about data management challenges. Scrum masters may find it difficult to facilitate effective daily stand-ups. They should ensure accurate and timely data collection, address data management challenges promptly, and facilitate open discussions to resolve data-related challenges.

Retrospectives: Retrospectives aims to reflect on the past iteration to understand what went well, what went wrong, and what could be improved [52, 19]. Data management challenges (e.g., challenges in managing the integration process, such as integrating development project data into retrospective agendas) can hinder the agile development team's ability to analyse project performance data, derive actionable insights, and implement improvements. Agile teams should use data integration tools, ensure data accuracy and completeness, and facilitate collaborative analysis sessions. Moreover, scrum masters may struggle to guide the team in reflecting on past iterations and deriving actionable insights. They should use structured retrospective techniques, ensure the availability of accurate data, and guide the team in data analysis. Product owners may face challenges in understanding the overall project performance and making informed decisions. Product owners and other agile team members should use comprehensive data analytics tools, ensure regular data reviews, and involve stakeholders in the retrospective process.
Testing and Quality Assurance: Data quality challenges necessitate better collaboration between testing and cross-functional agile teams to ensure data integrity throughout the agile development lifecycle. Software testers should implement automated data quality checks, involve QA early in the data collection process, and ensure continuous collaboration with agile development teams. To ensure data accuracy, subject-matter experts should be involved early in the data collection and integration phases.

As the industry continues to evolve towards more data-driven decision-making [57, 2], the insights from the SLR and the survey can impact how practitioners approach data management in agile software development environments, leading to the adoption of new strategies that enhance agility, improve product quality, and facilitate better project outcomes.

6.3 Recommendations for Practitioners

We recommend developing comprehensive data management policies to address the data management challenges in support of agile methodologies used by practitioners. The data management policies require a collaborative effort among multiple stakeholders. A data governance team can lead the policy creation, ensuring alignment with regulatory requirements and organisational goals. Our

SLR and survey show that including roles in the agile development team, such as legal advisors, can guide legal obligations and help the agile development team comply with data privacy regulations [15]. The challenges that pertain to large volumes of data (e.g., ensuring data quality of large datasets) can be handled by developers, who also help ensure that machine learning models are trained on accurate and representative datasets [59]. Agile coaches assist teams in implementing data-driven decision-making policies, which help effectively utilise data to guide development processes [9]. To develop comprehensive data management policies, we recommend that agile teams review the data management challenges and pick suitable solutions that suit their project and agile environment settings.

Software Engineering techniques are now widely applied to Machine Learning (ML) and data-intensive projects [5], helping with tasks like managing datasets, ensuring reproducibility, and keeping track of model versions. SE4ML (Software Engineering for Machine Learning) [69] uses structured software practices to handle these needs, which is especially helpful for agile projects with changing data requirements. Likewise, AI-driven methods in Software Engineering (AI4SE) address complex data management challenges [43]. These approaches can help agile teams to consistently manage evolving data and adapt more smoothly to changes across project iterations.

While significant in agile environments, the data management challenges identified in this thesis are not necessarily exclusive to agile projects. Many of these issues—such as data integration, quality control, and versioning—are universal to software engineering as a discipline. However, agile's iterative and adaptive nature can amplify these challenges due to the need for continuous updates, rapid feedback loops, and real-time data alignment. For example, agile projects often face complexities in maintaining data quality and consistency through frequent development cycles, as discussed in Sections 4.7 and 5.3. In contrast, a more structured methodology may encounter fewer interruptions in data management, as changes are typically consolidated at specific checkpoints. Nonetheless, while structured methodologies can benefit from more predictability, they may also lack the responsiveness needed for real-time data requirements, which agile can better accommodate. This thesis primarily focuses on agile because it uniquely intensifies data management challenges. However, it acknowledges that many of these challenges are shared across software engineer-

ing practices and could benefit from broader data management strategies that transcend specific methodologies.

Another factor to consider in the data management challenges identified in this thesis is the concept of Agile Theater [70], where teams perform agile ceremonies and use agile terminology without fully embracing the underlying principles. This phenomenon can result in practices that appear agile but lack the adaptability, collaboration, and iterative improvement central to true agile methodologies. In such cases, teams may face data management challenges due to incomplete or inconsistent implementation of agile practices. For example, if data is not continuously aligned with evolving project requirements—often due to rigid, siloed practices—this can result in issues with data integration, quality, and timely availability. While this thesis assumes that participants were, to some extent, appropriately employing agile practices, the presence of *agile theatre* could partially explain certain challenges observed in Sections 4.7 and 5.3. Addressing those issues would require further investigation to determine if a lack of full agile adoption contributes to the data management issues, suggesting that some challenges may stem from an incomplete agile transformation rather than intrinsic flaws in the agile methodology itself. This is an interesting area for future research.

6.4 Summary

This chapter discussed the results of the SLR and the practitioner survey. We compared the knowledge gained from the literature review with practitioners' experiences across various industries. Our discussion highlighted several data management challenges, including managing data integration processes, ensuring data accuracy and consistency, comprehensively collecting data, and managing complex data analysis. The practitioner survey insights confirm the majority of the data management challenges from the SLR. Additionally, the discussion presented the potential implications of these challenges on agile process activities and agile project roles. The discussion emphasised the need for enhanced data management strategies, advanced technologies, and training to improve project outcomes. Furthermore, recommendations were provided to help agile teams address these challenges. Overall, the chapter highlighted the importance of efficient data management in ensuring the success of agile projects. Next, we

present the threats to validity in Chapter 7

Chapter 7

Threats to Validity

In this chapter, we investigate potential validity threats to the findings of this thesis. We explore aspects that could compromise *internal*, *external*, *construct*, and *conclusion* validity. Careful steps were taken throughout the design and execution of this thesis to address these threats [71], which are discussed in detail below.

7.1 Internal Validity

Search bias is a potential threat to our SLR because the search strategy unintentionally might favour some studies or exclude relevant ones. It is possible that a search bias could make the SLR findings less complete and reliable. Because of this bias, we might not fully understand the challenges and solutions associated with data management in agile software development, which could lead to incorrect or incomplete findings.

To mitigate search biases, we implemented a comprehensive search strategy that included iterative pilot searches to refine search terms, ensuring that our search strings were broad enough to capture all relevant literature while being specific enough to exclude irrelevant studies (see Section 3.1.1). We also defined and consistently applied inclusion and exclusion criteria during the study selection process (see Section 3.1.2). We also conducted thorough quality checks using a predetermined quality assessment framework with at least one additional reviewer involved, resolving any disagreements through discussion [28].

There is a possibility that *selection bias* will happen when studies are chosen to be included in the SLR. This could result in a sample that is not representative or skewed, meaning that some categories might be over- or under-represented. Such bias can make the thesis's conclusions less reliable.

To mitigate selection bias, we continue using the same criteria to decide which studies to include or exclude (see Section 3.1.2). This process helped ensure that the SLR only included studies that met certain relevant criteria.

When evaluating the selected studies, there is a chance that *quality assessment bias* will occur. Personal opinions could make the quality ratings less consistent and dependable. Quality assessment bias could cause lower-quality studies to be included or higher-quality studies to be left out, which could influence the overall results of the research.

We carefully checked the quality of each study we chose to make sure it was rigorous and relevant. In order to accomplish this, each study was assessed using a predefined quality assessment framework (see Section 3.1.5). Any disagreements were then resolved through discussion, with at least one additional reviewer involved.

The possibility of *data extraction bias* exists when data is extracted from the chosen studies, leading to the possible omission or incorrect interpretation of some data. Data extraction bias could cause incomplete or incorrect data to be used in the analysis, making the thesis's results less reliable.

We used a detailed data extraction procedure to mitigate data extraction bias. The initial data extraction was followed by verification through meetings with at least one other collaborator. We followed established guidelines for thematic classification [36] and employed well-established methods for quantitative and qualitative data analysis [28].

For the survey, a *design bias* may occur if the questions are not relevant, clear, or aligned with the research's objectives. If the survey questions are biased or poorly designed, the data collected may not accurately reflect the practitioners' experiences, leading to unreliable conclusions.

To address this threat, we iteratively designed the survey questions in collaboration with multiple reviewers (collaborators) to ensure relevance and reliability (see Section 3.2.1). The questionnaire was piloted with three practitioners experienced in data management issues, and their feedback was used to refine the questions. Additionally, we consulted an experienced external researcher with

a strong background in survey studies to review the questionnaire and provide feedback on its clarity and relevance before it was piloted with practitioners.

Bias may occur during the data analysis of the practitioner survey, mainly through *confirmation bias* or *researcher bias*. If biases are not carefully addressed during data analysis, the results could be skewed, leading to inaccurate or misleading conclusions.

To minimise bias during data analysis, we conducted discussion sessions to cross-check data interpretations. These sessions involved discussing the development of the coding scheme, the classification of themes, and the interpretation of ambiguous responses. The outcomes were reviewed with at least one additional reviewer to reach a final decision, ensuring consistency and reliability in the analysis.

7.2 External Validity

Possible *threat to generalisability* might occur due to the relatively small survey sample or the SLR's scope, which might not accurately reflect the larger context of data management in agile software development. This indicates that the research may not apply to other agile software development environments if the results cannot be applied to a wide range of settings.

To enhance the generalizability of SLR, we utilised the wide-ranging coverage provided by the Scopus database, which includes work from all relevant software engineering publication venues, ensuring a broad and representative sample of studies [28]. We also followed a strict search protocol, which guided the systematic data extraction and review. While our findings may not be universally generalisable, they apply to various agile software development settings. Regarding the generalisability of the practitioner survey results, we aimed to target a diverse group of practitioners from various industries and levels of experience in agile software development to reduce possible bias from certain groups.

7.3 Construct Validity

Data extraction bias is a potential threat to both the SLR and the survey, where inconsistencies in extracting data could lead to inaccurate or incomplete

findings. Such bias can compromise the accuracy and reliability of the thesis's results, reducing the overall validity of the research.

We implemented standardised data extraction forms to reduce data extraction bias in the SLR, ensuring that we applied consistent criteria and procedures across all studies. Furthermore, we clearly defined the data management terms such as "data integration" and "data quality", derived from recognised research in the field,to maintain consistency in how we interpreted and analysed these concepts throughout the thesis. Additionally, our data extraction method was refined after several pilot tests to enhance accuracy. We reviewed the extracted data with at least one additional reviewer (a collaborator), resolving any discrepancies through consensus. Each study's extracted data was then discussed with the reviewers to ensure consistency. We maintained a detailed description of the extraction process to ensure transparency and allow other researchers to verify our methods and findings. To mitigate data extraction bias in the survey, we followed a standardised process in Section 3.2.2 for analysing responses and had multiple reviewers check and agree on the data extracted. Moreover, we tested the survey first to maintain the questions were clear, and we used tool NVivo (version 14)[1] to help organise responses consistently.

Questionnaire Construct validity is at risk if the survey questions do not correctly measure what they are supposed to, which can lead to unreliable or incorrect results. If the survey does not measure what it should, the results might not show what participants really think, leading to wrong conclusions and weakening the thesis's reliability.

The survey questionnaire was designed to accurately measure the intended constructs [72, 39, 73]. We considered convergent and divergent validity and integrated these into our survey design. We achieved this through the execution of a pilot survey, which helped to ensure the construct and content validity. We also conducted inter-coder reliability checks and peer reviews with the collaborators to enhance the validity of the thematic analysis. An iterative process continuously improved the codes and themes, ensuring they were firmly grounded in the data and accurately reflected the participants' perspectives. Details of the thematic analysis process are explained in Section 3.2.2.

[1] `https://lumivero.com/products/nvivo/`

7.4 Conclusion Validity

It is possible that the conclusions drawn from the data might not be correct or fully supported by the evidence, leading to incorrect or misleading interpretations. If the conclusion validity is compromised, the thesis's results may not be reliable or valid. This can lead to incorrect interpretations of the data, such as overestimating or underestimating the impact of certain variables (e.g., data integration challenges on project success). This can threaten the overall contributions of the research and its applicability to agile software settings.

To mitigate conclusion validity, we conducted a cross-validation process in which at least two reviewers analysed the extracted data to cross-validate findings and interpretations, minimising the likelihood of inaccurate conclusions (from both the SLR and the practitioner survey). An iterative review process involving peer discussions and revisions based on collective feedback helped to strengthen the reliability of our conclusions. We took precautions to ensure that our conclusions from the thesis findings are reliable. We attempted to build a representative sample size to identify meaningful patterns in the data, as determined by a power analysis. In addition, we employed robust statistical techniques to examine the data, which involved verifying the consistency of our findings through cross-validation. This assisted us in avoiding common mistakes and ensuring that our discoveries, such as the impact of data integration challenges on project success, were reliable and transparent. By implementing these measures, we desired to mitigate any uncertainties regarding the validity of the findings in our thesis.

Chapter 8

Conclusion

Agile development teams face various challenges when managing data related to software products and processes. These challenges include combining data from various sources, maintaining data integrity in the face of constant change, and adjusting to agile methods' dynamic nature. Even though data management is essential to agile software development, there has been a lack of understanding of the challenges or a methodical way to solve them.

The main aim of this thesis is to systematically explore the challenges related to data management in agile software development and uncover potential solutions for these identified challenges. To that end, the first conducted a systematic literature review by reviewing 45 research papers in order to identify and classify various aspects of data management as well as the challenges and solutions related to them. The review employed a thorough search strategy using the Scopus database. The obtained data was synthesised through the use of thematic analysis. In order to gain practical perspectives from agile professionals in different roles, we carried out a practitioner survey with 32 industry practitioners. In order to ensure thorough coverage of relevant challenges and solutions, the SLR findings were considered during the survey design process. We employed qualitative and quantitative analysis techniques to analyse and interpret the survey data. To ensure the validity of the findings, we followed a rigorous methodology consisting of a thorough search strategy, consistent criteria application, and cross-validation techniques. We discuss our observations and findings below.

8.1 Observations and Contributions

The research used the SLR and a practitioner survey to identify several important data management challenges in agile software development. One of the main challenges we identified is managing data integration processes. Capturing diverse data and automating data collection also presents a significant challenge for agile teams. Additionally, ensuring data accuracy, consistency, and completeness are critical challenges. The complexity of data and the need for real-time analysis add further challenges.

The results of the SLR showed that challenges with integrating data from various sources, preserving data accuracy, and handling inconsistent or incomplete data were common. The survey supported these results, which also revealed more insights from agile practitioners, such as the need for thorough training programs that emphasise data management skills and the difficulties in automating data collection. Proposed solutions to enhance data integration, quality, and analysis include the use of cloud-based platforms, automated tools, ontology-based approaches, and communication-centric methodologies. The results highlight the need for advanced tools and techniques, training, and effective data management policies in order to improve project success and decision-making in agile settings.

This thesis makes a substantial contribution to the body of knowledge on data management in agile software development. Through the SLR, the thesis systematically categorised the challenges and their solutions. The practical insights from practitioners were used to validate the findings, leading to a comprehensive understanding of data management challenges in agile settings. This thesis emphasises the importance of incorporating advanced analytical methods and automated tools into agile processes. It also emphasises how important it is to have focused training programs to improve agile teams' capacity for data management. In order to address new data management challenges in agile software development, it also identifies gaps in current practices. It proposes future research directions, such as creating ontology-based solutions and real-time analytics tools.

Prior research has examined particular data management challenges within particular projects and organisations. On the other hand, this thesis offers a more comprehensive viewpoint by methodically examining a range of software

development projects from different industries and integrating findings from a practitioner survey. The integrated methodological approach highlights the interconnectedness of data integration, quality, collection, and analysis and ensures a thorough understanding of data management challenges and solutions. The thesis provides a more comprehensive view and recommends future research directions.

8.2 Implications and Recommendations

Data management challenges in agile software development significantly impact agile activities (requirement gathering, sprint planning, and daily stand-ups), all of which are essential for the success of agile projects. Inaccurate or lacking data during requirements gathering can result in poorly defined project goals and user stories, which will have a detrimental effect on development and stakeholder satisfaction. Effective communication and comprehensive data collection techniques are crucial to reducing this. Data quality challenges in sprint planning can lead to poorly stated sprint objectives and problems with task prioritisation, which frequently call for spike sprints. Scrum masters are advised to have strong data validation procedures and to communicate openly about any data challenges. Incomplete data can impede daily stand-ups, making it more difficult to identify roadblocks and make decisions. Open communication regarding data challenges and real-time data tracking tools are essential. The research highlights the significance of formulating specific guidelines for data management and offering team education to effectively tackle issues related to data integration, quality, and analysis. The article emphasises the efficacy of automation tools, decentralised data management practices, and the incorporation of machine learning developers and legal advisors into agile teams to improve data management and regulatory compliance.

8.3 Future Research Directions

Future studies should investigate advanced integration techniques to harmonise and interoperate diverse and heterogeneous data sources in agile frameworks. Promising directions for research are ontology-based solutions and architecture-centric approaches. Research should focus on systematic and automated data

collection methodologies to guarantee accuracy and completeness throughout the agile software development life-cycle. Advanced automated quality assurance techniques should be developed, combining real-time quality metrics, dashboards, and continuous monitoring tools. Advanced real-time analytics techniques and tools that can manage large volumes of both structured and unstructured data should be the focus of future research in order to improve the decision-making abilities of agile teams. Solid data privacy and security frameworks that can be seamlessly included in agile processes, including automated compliance checks and encryption techniques, need more research to be developed. Furthermore, researching the impact of specialised training programs on enhancing data management proficiency among agile team members, explicitly emphasising data integration, quality assurance, and analytics in the agile framework. Future research should include rigorous evaluation of the proposed solutions from both the SLR and practitioners' surveys to assess their effectiveness and applicability in real-world agile environments.

Bibliography

[1] A. Fawzy, A. Tahir, M. Galster, and P. Liang, "Exploring data management challenges and solutions in agile software development: A literature review and practitioner survey," *arXiv preprint arXiv:2402.00462*, 2024.

[2] C. Matthies and G. Hesse, "Towards using data to inform decisions in agile software development: views of available data," *arXiv preprint arXiv:1907.12959*, 2019.

[3] "What is data management?." `https://www.ibm.com/topics/data-management`. Accessed: 2023-12-07.

[4] S. Ambler, "When it gets cultural: Data management and agile development," *IT Professional*, vol. 10, no. 6, pp. 11–14, 2008.

[5] U. M. Graetsch, H. Khalajzadeh, M. Shahin, R. Hoda, and J. Grundy, "Dealing with data challenges when delivering data-intensive software solutions," *IEEE Transactions on Software Engineering*, 2023.

[6] X. Franch, L. Lopez, S. Martínez-Fernández, M. Oriol, P. Rodríguez, and A. Trendowicz, "Quality-aware rapid software development project: the q-rapids project," in *Proceedings of the 51st International Conference on Software Technology: Methods and Tools (TOOLS)*, pp. 378–392, Springer, 2019.

[7] D. Larson and V. Chang, "A review and future direction of agile, business intelligence, analytics and data science," *International Journal of Information Management*, vol. 36, no. 5, pp. 700–710, 2016.

[8] A. Fabijan, H. H. Olsson, and J. Bosch, "The lack of sharing of customer data in large software organizations: challenges and implications," in *Proceedings of the 17th International Conference on Agile Software Development (XP)*, pp. 39–52, Springer, 2016.

[9] F. A. Batarseh and A. J. Gonzalez, "Predicting failures in agile software development through data analytics," *Software Quality Journal*, vol. 26, pp. 49–66, 2018.

[10] W. N. Behutiye, P. Rodríguez, M. Oivo, and A. Tosun, "Analyzing the concept of technical debt in the context of agile software development: A systematic literature review," *Information and Software Technology*, vol. 82, pp. 139–158, 2017.

[11] E. D. Canedo, A. T. S. Calazans, I. N. Bandeira, P. H. T. Costa, and E. T. S. Masson, "Guidelines adopted by agile teams in privacy requirements elicitation after the brazilian general data protection law (lgpd) implementation," *Requirements Engineering*, vol. 27, no. 4, pp. 545–567, 2022.

[12] G. Lee and W. Xia, "Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility," *MIS Quarterly*, vol. 34, no. 1, pp. 87–114, 2010.

[13] A. Harriman, P. Hodgetts, and M. Leo, "Emergent database design: liberating database development with agile practices," in *Proceedings of the 5th International Conference on Agile Software Development (XP)*, pp. 100–105, IEEE, 2004.

[14] K. Vestues, G. K. Hanssen, M. Mikalsen, T. A. Buan, and K. Conboy, "Agile data management in nav: a case study," in *Proceedings of the 23rd International Conference on Agile Software Development (XP)*, pp. 220–235, Springer, 2022.

[15] A. Barbala, T. Sporsem, and V. Stray, "Data-driven development in public sector: How agile product teams maneuver data privacy regulations," in *Proceedings of the 24th International Conference on Agile Software Development (XP)*, pp. 165–180, Springer, 2023.

[16] H.-M. Chen, R. Kazman, and S. Haziyev, "Agile big data analytics development: An architecture-centric approach," in *Proceedings of the 49th*

Hawaii International Conference on System Sciences (HICSS), pp. 5378–5387, IEEE, 2016.

[17] C. Rosenkranz, R. Holten, M. Räkers, and W. Behrmann, "Supporting the design of data integration requirements during the development of data warehouses: a communication theory-based approach," *European Journal of Information Systems*, vol. 26, pp. 84–115, 2017.

[18] J. A. Pater, S. Lie-Tjauw, M. Gonzalez, M. Kim, S. Isbell, and D. Severson, "Advancing the agile software process: The case of modernizing the army community service's information technology infrastructure," 2018.

[19] C. Matthies, "Playing with your project data in scrum retrospectives," in *Proceedings of the 42nd ACM/IEEE International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pp. 113–115, IEEE, 2020.

[20] J. Lin, H. Yu, Z. Pan, Z. Shen, and L. Cui, "Towards data-driven software engineering skills assessment," *International Journal of Crowd Science*, vol. 2, no. 2, pp. 123–135, 2018.

[21] S. Kaur, S. Siarov, M. Witzmann, and L. Lindblad, "A dialogue on the digitization of requirements, verification, and test management with data-driven systems engineering (ddse)," in *Proceedings of the 71st International Astronautical Congress (IAC)*, pp. 1–6, IAF, 2020.

[22] F. Rahman and P. Devanbu, "How, and why, process metrics are better," in *Proceedings of the 35th International Conference on Software Engineering (ICSE)*, pp. 432–441, IEEE, 2013.

[23] W. Li, "Software product metrics," *IEEE Potentials*, vol. 18, no. 5, pp. 24–27, 1999.

[24] F. N. Colakoglu, A. Yazici, and A. Mishra, "Software product quality metrics: A systematic mapping study," *IEEE Access*, vol. 9, pp. 44647–44670, 2021.

[25] E. Grünewald, "Cloud native privacy engineering through devprivops," in *Proceedings of the IFIP International Summer School on Privacy and Identity Management (IFIP)*, pp. 122–141, Springer, 2021.

[26] Z. Min, L. Qiong-mei, and W. Cheng, "Practices of agile manufacturing enterprise data security and software protection," in *Proceedings of the 2nd International Conference on Industrial Mechatronics and Automation (ICIMA)*, vol. 1, pp. 318–321, IEEE, 2010.

[27] M. P. Barcellos, "Towards a framework for continuous software engineering," in *Proceedings of the XXXIV Brazilian Symposium on Software Engineering (SBES)*, pp. 626–631, ACM, 2020.

[28] B. Kitchenham, S. Charters, *et al.*, "Guidelines for performing systematic literature reviews in software engineering version 2.3," *Engineering*, vol. 45, no. 4ve, p. 1051, 2007.

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

[30] A. S. Campanelli and F. S. Parreiras, "Agile methods tailoring–a systematic literature review," *Journal of Systems and Software*, vol. 110, pp. 85–100, 2015.

[31] E. Mourão, J. F. Pimentel, L. Murta, M. Kalinowski, E. Mendes, and C. Wohlin, "On the performance of hybrid search strategies for systematic literature reviews in software engineering," *Information and Software Technology*, vol. 123, p. 106294, 2020.

[32] A. Carrera-Rivera, W. Ochoa, F. Larrinaga, and G. Lasa, "How-to conduct a systematic literature review: A quick guide for computer science research," *MethodsX*, vol. 9, p. 101895, 2022.

[33] H. Zhang, M. A. Babar, and P. Tell, "Identifying relevant studies in software engineering," *Information and Software Technology*, vol. 53, no. 6, pp. 625–637, 2011.

[34] A. Fawzy, A. Tahir, M. Galster, and P. Liang, "Dataset of the paper "exploring data management challenges and solutions in agile software development: A literature review and practitioner survey"." `https://doi.org/10.5281/zenodo.10597817`, 2024.

[35] K. Krippendorff, *Content analysis: An introduction to its methodology*. Sage publications, 2018.

[36] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, p. 77, 2006.

[37] "Study quality assessment tools." `https://www.nhlbi.nih.gov/health-topics/study-quality-assessment-tools`. Accessed: 2023-12-21.

[38] T. Punter, M. Ciolkowski, B. Freimut, and I. John, "Conducting on-line surveys in software engineering," in *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings.*, pp. 80–88, IEEE, 2003.

[39] Linåker, Johan and Sulaman, Sardar Muhammad and Maiani de Mello, Rafael and Höst, Martin, "Guidelines for Conducting Surveys in Software Engineering," tech. rep., Department of Computer Science, Lund University, 2015.

[40] M. Q. Patton, *Qualitative research & evaluation methods: Integrating theory and practice*. Sage publications, 2014.

[41] K. E. Harper and A. Dagnino, "Agile software architecture in advanced data analytics," in *Proceedings of the 11th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pp. 243–246, IEEE, 2014.

[42] P. S. dos Santos Júnior, M. P. Barcellos, and J. P. A. Almeida, "An ontology-based approach to enable data-driven decision-making in agile software organizations," in *Proceedings of the XIV Seminar on Ontology Research in Brazil (ONTOBRAS)*, pp. 279–284, 2021.

[43] D. Chhillar and K. Sharma, "Act testbot and 4s quality metrics in xaas framework," in *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 503–509, IEEE, 2019.

[44] B. Upender, "Staying agile in government software projects," in *Proceedings of the 6th International Conference on Agile Software Development (XP)*, pp. 153–159, IEEE, 2005.

[45] A. A. Abdallah and I.-S. Fan, "Towards building ontology-based applications for integrating heterogeneous aircraft maintenance records," in *Proceedings of the 20th IEEE International Conference on Industrial Informatics (INDIN)*, pp. 293–299, IEEE, 2022.

[46] M. Rix, B. Kujat, T. Meisen, and S. Jeschke, "An agile information processing framework for high pressure die casting applications in modern manufacturing systems," *Procedia CIRP*, vol. 41, pp. 1084–1089, 2016.

[47] H. Spengler, C. Lang, T. Mahapatra, I. Gatz, K. A. Kuhn, F. Prasser, *et al.*, "Enabling agile clinical and translational data warehousing: Platform development and evaluation," *JMIR Medical Informatics*, vol. 8, no. 7, p. e15918, 2020.

[48] V. Kannan, J. S. Fish, J. M. Mutz, A. R. Carrington, K. Lai, L. S. Davis, J. E. Youngblood, M. R. Rauschuber, K. A. Flores, E. J. Sara, *et al.*, "Rapid development of specialty population registries and quality measures from electronic health record data," *Methods of Information in Medicine*, vol. 56, no. S01, pp. e74–e83, 2017.

[49] M. Hofer, S. Hellmann, M. Dojchinovski, and J. Frey, "The new dbpedia release cycle: Increasing agility and efficiency in knowledge extraction workflows," in *Proceedings of the 16th International Conference on Semantic Systems (SEMANTiCS)*, pp. 1–18, Springer, 2020.

[50] T. Little, F. Greene, T. Phillips, R. Pilger, and R. Poldervaart, "Adaptive agility," in *Proceedings of the 5th International Conference on Agile Software Development (XP)*, pp. 63–70, IEEE, 2004.

[51] C. Schüttler, H.-U. Prokosch, M. Hummel, M. Lablans, B. Kroll, C. Engels, and G. B. A. I. development team, "The journey to establishing an it-infrastructure within the german biobank alliance," *Plos One*, vol. 16, no. 9, p. e0257632, 2021.

[52] S. R. Dharmapal and K. T. Sikamani, "Big data analytics using agile model," in *Procedings of the International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 1088–1091, IEEE, 2016.

[53] S. Dursun, K. Duman, T. Tuna, M. Abbas, and J. Ding, "A workflow for intelligent data-driven analytics software development in oil and gas industry," in *Proceedings of the SPE Annual Technical Conference and Exhibition (ATCE)*, pp. SPE–170859, SPE, 2014.

[54] V. Vøgt, J.-A. Harrs, V. Reinhart, P. Hollenbach, M. M. B ̆hler, and T. Tewes, "Implementing agile data workflows to unlock climate-resilient urban planning," *Climate*, vol. 11, no. 9, p. 174, 2023.

[55] H. Huang, Y. He, L. Zhang, Z. Zeng, T. Ouyang, and Z. Zeng, "Leveraging modern big data stack for swift development of insights into social developments," in *Proceedings of the International Conference on Wireless Communications, Networking and Applications (WCNA)*, pp. 325–333, Springer, 2021.

[56] S. Martínez-Fernández, A. M. Vollmer, A. Jedlitschka, X. Franch, L. López, P. Ram, P. Rodríguez, S. Aaramaa, A. Bagnato, M. Choraś, *et al.*, "Continuously assessing and improving software quality with software analytics tools: a case study," *IEEE Access*, vol. 7, pp. 68219–68239, 2019.

[57] R. B. Svensson, R. Feldt, and R. Torkar, "The unfulfilled potential of data-driven decision making in agile software development," in *Proceedings of the 20th International Conference on Agile Software Development (XP)*, pp. 69–85, Springer, 2019.

[58] T. Lehtonen, T. Aho, K. Kuusinen, and T. Mikkonen, "Visualizations for software development process management," *Information Modelling and Knowledge Bases XXVIII*, vol. 292, p. 1, 2017.

[59] R. Dautov, E. J. Husom, and F. Gonidis, "Towards mlops in mobile development with a plug-in architecture for data analytics," in *Proceedings of the 6th International Conference on Computer, Software and Modeling (ICCSM)*, pp. 22–27, IEEE, 2022.

[60] M. Das, R. Cui, D. R. Campbell, G. Agrawal, and R. Ramnath, "Towards methods for systematic research on big data," in *Proceedings of the IEEE International Conference on Big Data (BigData)*, pp. 2072–2081, IEEE, 2015.

[61] H. H. Olsson, "Challenges and strategies for undertaking continuous experimentation to embedded systems: Industry and research perspectives," *Agile Processes in Software Engineering and Extreme Programming*, vol. 277, 2018.

[62] C. Fagarasan, C. Cristea, M. Cristea, O. Popa, and A. Pisla, "Integrating sustainability metrics into project and portfolio performance assessment in agile software development: A data-driven scoring model," *Sustainability*, vol. 15, no. 17, p. 13139, 2023.

[63] S. Hamer, C. Quesada-López, and M. Jenkins, "Students' perceptions of integrating a contribution measurement tool in software engineering projects," in *Proceedings of the 35th IEEE International Conference on Software Engineering Education and Training (CSEE&T)*, pp. 21–30, IEEE, 2023.

[64] J. Bosch, "Towards a digital business operating system," in *Proceedings of the 13th International Conference on Research Challenges in Information Science (RCIS)*, pp. 1–9, IEEE, 2019.

[65] B. Alsaadi and K. Saeedi, "Data-driven effort estimation techniques of agile user stories: a systematic literature review," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5485–5516, 2022.

[66] T. Jenness, F. Economou, K. Findeisen, F. Hernandez, J. Hoblitt, K. S. Krughoff, K. Lim, R. H. Lupton, F. Mueller, W. O'Mullane, *et al.*, "Lsst data management software development practices and tools," in *Proceedings of the Software and Cyberinfrastructure for Astronomy V (SPIE)*, pp. 50–69, SPIE, 2018.

[67] S. Chung and E. Hartford, "Bridging the gap between data models and implementations: Xmi2sql," in *Proceedings of the International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT-ICIW)*, pp. 201–201, IEEE, 2006.

[68] H. Al Hashimi and A. Gravell, "Spikes in agile software development: An empirical study," in *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 1715–1721, IEEE, 2020.

[69] G. Giray, "A software engineering perspective on engineering machine learning systems: State of the art and challenges," *Journal of Systems and Software*, vol. 180, p. 111031, 2021.

[70] W. Felidré, L. Furtado, D. A. Da Costa, B. Cartaxo, and G. Pinto, "Continuous integration theater," in *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 1–10, IEEE, 2019.

[71] A. Ampatzoglou, S. Bibi, P. Avgeriou, and A. Chatzigeorgiou, "Guidelines for managing threats to validity of secondary studies in software engineering," *Contemporary empirical methods in software engineering*, pp. 415–441, 2020.

[72] B. A. Kitchenham and S. L. Pfleeger, "Personal opinion surveys," in *Guide to advanced empirical software engineering*, pp. 63–92, Springer, 2008.

[73] C. Robson, "Real world research: A resource for social scientists and practitioner-researchers," *(No Title)*, 2002.

Appendices

Appendix A:
Detailed Description of Data Management Challenges

Table A1: Data Integration Challenges

Challenges	**Description and Example**
Data Harmonisation and Interoperability	Ensuring that data from different sources can be understood and utilised together by aligning common formats and schemas without altering the original data content. Example: Converting date formats from MM/DD/YYYY in one system to YYYY-MM-DD in another to unify sales data analysis.
Semantic Heterogeneity	Different meanings or interpretations of the same data across various systems. Example: Aligning differing patient discharge status terms, like "Released" vs. "Discharged," across multiple hospital systems.
Data Transformation and Extraction	Modifying data's structure, format, or content to meet the needs of a new application or system, often involving significant changes. Example: Changing product sizes from "S, M, L, XL" to detailed measurements (e.g., "Small (S) - Chest: 34-36 inches") and converting inventory data into JSON for a modern system.

Continued on next page

Table A1: Data Integration Challenges (Continued)

Challenges	**Description and Example**
Managing Data Integration	Coordinating the process required to merge data from disparate sources into a unified system. Example: Integrating customer data from online banking, call centres, and branch visits into a single data warehouse, requiring a unified data model because each of these departments uses different systems and standards
Complexity of Integrating Real-Time Data	Merging and processing data that is continuously generated from various sources, requiring immediate processing and integration for timely decision-making. Example: Integrating live stock market feeds, transaction data from payment systems, and customer feedback for real-time analytics in a financial app.

Table A2: Data Collection Challenges

Challenges	**Description and Example**
Capturing Diverse Data	The challenge of collecting structured and unstructured data from varied sources. Example: Analyzing customer sentiment requires collecting purchase data and social media comments.
Data Collection Method	Difficulties arising from using different data gathering tools or techniques. Example: An agile development team faces challenges with inconsistent data collection from device sensors across various smartphones. Differences in sensor accuracy among smartphones result in gaps in collecting user activity data (what users are doing on their smartphones).
Informative Data Collection	Ensuring that the collected data provides the insights needed for decision-making. Example: User activity data is collected, but lacks details on specific user experience issues.
Comprehensive Data Collection	Systematically collecting data across all project aspects. Example: An e-commerce team might miss out on user experience and satisfaction data while focusing on backend metrics.
Automation Challenges in Data Collection	Implementing automated systems for data collection efficiently and accurately. Example: An automated error logging system incorrectly categorises user-generated errors, complicating bug prioritisation.

Table A3: Data Quality Challenges

Challenges	**Description and Example**
Ensuring Data Accuracy and Consistency Across Sources	Ensuring data from different sources is accurate and presented in a uniform format. Example: Integrating stock market data reveals inconsistencies in trading volume reports from different exchanges, requiring data normalisation for accurate application reporting.
Completeness of Data	Ensuring every necessary data field is filled without missing critical information. Example: User stories for a new feature are found to be missing acceptance criteria during a sprint review, leading to project delays and rework.
Effective Data Quality Management	Implementing continuous practices to uphold data integrity and address issues promptly. Example: Data corruption challenges overwhelm automated correction efforts, making manual reviews necessary and impacting development timelines.
Data Quality Standardisation	Establishing and applying uniform measures for data quality assessment across varied projects. Example: A team faces difficulty creating data quality metrics that are relevant and applicable to all projects, hindering company-wide quality assessments.

Table A4: Data Analysis Challenges

Challenges	**Description and Example**
Complex Data Sets	Challenges posed by analysing large or complex datasets. Example: A team refines vast user data over many sprints for to build a machine learning project.
Real-Time Analysis Requirements	The challenge of analysing data as it is generated, without delay, to support immediate decision-making or operational actions. Example: Developing a real-time user engagement tracker, frequently revisiting their approach to meet analysis speed requirements.
Analyzing Semantic Heterogeneity Data	Difficulties with data with varying meanings across sources. Example: A healthcare team struggles to align medical procedure terms across datasets.
Ensuring Analytical Accuracy	Keeping data analysis precise and reliable to minimise errors. Example: A financial project team uses pair programming to boost analysis accuracy, adding unforeseen tasks to their agile process.
Selection of Appropriate Analytical Techniques	Choosing suitable analysis methods based on data characteristics. Example: Debating over using statistical models or machine learning for developing a fraud detection system

Appendix B:
Detailed Description of Data Management Solutions

Table B1: Data Integration Solutions

Solutions	**Description and Example**
Development of Ontologies	Using tools or guides to create a unified vocabulary for data across systems. Example: Employing an ontology editor like Protégé tool to build ontologies that standardise company-wide data terms.
Cloud-Based Platforms	Using cloud services for efficient data management and integration. Example: Utilising AWS for scalable data ingestion and transformation.
Communication-Centric Approaches	Applying communication theory to harmonise data meanings across teams. Example: Holding regular team meetings to ensure consistent data interpretation among stakeholders.
Automated Continuous Testing	Employing automated testing frameworks. Example: Using Jenkins, Travis CI, or GitHub Actions to regularly check data quality and integrity as part of the development process.

Continued on next page

Table B1: Data Integration Solutions (Continued)

Solutions	**Description and Example**
Agile Workflow Adaptations	Adjusting development practices to improve data integration outcomes. Example: integrating data quality checks into daily stand-ups, adjusting sprint reviews to include data integration retrospectives, or adopting pair programming for complex integration tasks.

Table B2: Data Collection Solutions

Solutions	**Description and Example**
User-Centered Design Strategies	Tailoring data collection methods to be intuitive and user-focused. Example: Designing surveys and interaction trackers based on user behaviour studies.
Automated Continuous Testing and Quality Metrics Dashboards	Monitoring data quality through automated tools and real-time dashboards. Example: Implementing continuous testing to quickly spot and rectify data issues.
Automation of Data Collection and Visualisation Toolchains	Using automated systems for efficient data gathering and visualisation. Example: Streamlining data workflows to enhance collection accuracy and analysis speed.

Continued on next page

Table B2: Data Collection Solutions (Continued)

Solutions	**Description and Example**
Centralised Data Management Systems	Centralizing data handling to improve consistency and data access. Example: Creating a unified data platform for all stages from collection to analysis.
Diagnostic Models for Data Collection and Sharing Practices	Assessing and refining data practices to improve quality and cooperation. Example: Developing models to optimise data gathering and sharing workflows.
Data-Driven Systems Engineering (DDSE) Methodologies	Integrating data analysis into system engineering for informed decisions. Example: Applying DDSE for collaborative and data-informed project development.
Legal Advisor for Data Sensitivity Assessment	Ensuring data collection complies with data protection laws. Example: Consulting legal experts to navigate privacy regulations in data practices.
Development of Ontology-Based Approaches	Enhancing data usefulness through structured data categorisation. Example: Using ontologies to standardise data terms for better interoperability.
Q-Rapids Tool for Valuable Information Acquisition	Incorporating tools designed for real-time data analysis in development. Example: Leveraging Q-Rapids for focused data collection in software projects.

Continued on next page

Table B2: Data Collection Solutions (Continued)

Solutions	**Description and Example**
Agile Methodology and Data Analyst Collaboration	Jointly developing data strategies with agile teams and data analysts. Example: Collaborating on adaptable data collection methods suited to project needs.
Participatory and Co-creative Workshops	Engage stakeholders in refining data collection and integration. Example: Hosting workshops to gather diverse inputs on improving data practices.

Table B3: Data Quality Solutions

Solutions	**Description and Example**
Comprehensive Data Quality Frameworks	Using established guidelines for ongoing data quality improvement. Example: Applying Total Data Quality Management (TDQM) principles to ensure data quality across the organisation.
Automated Data Cleaning and Enrichment Tools	Utilising software to automatically correct and enhance data. Example: Deploying Tools like Talend or Trifacta for real-time data cleaning and standardisation.
Data Governance Across Lifecycle	Creating rules and roles for data management from creation to deletion. Example: Establishing a governance committee to oversee data security and privacy.
Standardization of Data Quality Metrics and Processes	Setting uniform data quality standards and tracking mechanisms. Example: Developing a set of data quality KPIs (e.g., accuracy, completeness, consistency, timeliness) for data accuracy and consistency, monitored using tools like data quality scorecards.
Integration and Collection Practices for Quality Assurance	Embedding quality checks into data collection and integration workflows. Example: Incorporating validation s teps in ETL processes to ensure data integrity from the start.
Integration of Advanced Analytic Platforms	Adopting high-tech platforms for enhanced data management and quality. Example: Using Apache Kafka for seamless real-time data integration and quality analysis.

Table B4: Data Analysis Solutions

Solutions	**Description and Example**
Advanced Analytical Tools and Techniques	Employing Utilising high-level software for in-depth analysis of complex datasets. Example: Using R and Python for advanced data exploration and Tableau for visualisation.
Real-Time Data Processing Technologies	Using technologies that enable immediate analysis of live data streams. Example: Utilising big data frameworks for streaming and on-the-fly data processing enables organisations to act on insights without delay.
Frameworks and Methods for Semantic Integration	Applying organised methods to harmonise data meanings across sources. Example: Utilising RDF (Resource Description Framework) and SPARQL queries with ontologies for consistent data interpretation.
Quality Control Measures in Data Analysis	Establishing checks to maintain the precision of data analysis outcomes. Example: Integrating validation steps and cross-validation in analysis workflows to ensure accuracy.
Guidelines for Analytical Technique Selection	Creating criteria to assist in selecting suitable data analysis methods. Example: Developing a decision matrix to guide the choice between statistical and machine learning techniques based on data specifics.
Adoption of Machine Learning and AI for Data Analysis	Utilising AI and machine learning to enhance pattern recognition and prediction in data. Example: Leveraging AI and libraries such as TensorFlow and PyTorch for automated insights and predictive analysis.

Appendix C:
Data Types

Table C1: Data Types as Discussed in the Studies

Study	Process Data	Project Data	Product Data	Operational Data	Business Data
[16]	X		X		
[18]				X	X
[17]					X
[11]	X	X			
[41]			X	X	
[25]				X	
[57]				X	X
[59]				X	
[8]	X	X			
[53]				X	
[52]					X
[60]					X
[26]				X	
[14]					X
[15]				X	X
[45]				X	
[61]				X	
[46]				X	
[47]				X	
[48]				X	
[51]		X	X		
[49]	X		X		
[55]					X
[54]				X	X
[44]	X	X	X		
[64]				X	X

Continued on next page

Table C1: Data Types as Discussed in the Studies (Continued)

Study	**Process Data**	**Project Data**	**Product Data**	**Operational Data**	**Business Data**
[4]	X	X			
[13]	X		X		
[67]			X		
[42]	X		X		
[2]	X				X
[6]	X		X		
[43]	X		X	X	
[66]	X	X			
[9]	X	X			
[65]	X	X			
[21]	X	X			
[19]	X	X			
[56]	X	X			
[20]	X	X			
[50]		X			
[27]	X	X			
[58]	X	X			
[63]	X	X			
[62]	X	X			

Appendix D:
Survey Questionnaire

Demographics Information

1. Have you worked or are you currently working on a project that follows agile software development practices?
   Yes (mandatory to continue)

2. What is your age range?

   - Under 25
   - 25-34
   - 35-44
   - 45-54
   - 55-64
   - 65 and above
   - Prefer not to say

3. Which of the following best describes your gender?

   - Male
   - Female
   - Non-binary
   - Prefer not to say

4. How many years of professional experience do you have in software development?

   - Less than 1 year
   - 2-5 years
   - 6-10 years
   - More than 10 years
   - Prefer not to say

5. What is your current level of seniority in your profession in software development?

    - Entry-level
    - Mid-level
    - Senior-level
    - Executive
    - Other: _____

6. What is the highest level of education you have completed in a computing-related discipline?

    - Bachelor's degree
    - Master's degree or higher
    - Doctorate
    - Professional degrees
    - Degree not related to computing-related discipline
    - Prefer not to say
    - N/A

7. What is your role in agile software development projects? (Select all that apply.)

    - Scrum Master
    - Developer
    - Product Owner
    - Project Manager
    - Other: _____

8. In which industry sector do you primarily work?

    - ICT
    - Government administration, defense, public safety
    - Manufacturing
    - Education

- Healthcare
- Other: _____

9. What is the primary location of your employer or employers? E.g., NZ, Australia, etc. _____

Instruction: **Consider the agile software development projects in which you were significantly involved and had notable data management challenges. Please refer to these projects when answering the questions in the following sections.**

Data Integration Challenges and Solutions Questions

Open-ended Question: What data integration challenges have you encountered in your projects? _____

10. Which of the following data integration challenges have you encountered in your projects? (Select all that apply.)

    - **Data Harmonization and Interoperability:** Ensuring that data from different sources can be understood and utilised together by aligning common formats and schemas without altering the original data content. *Example: Converting date formats from MM/DD/YYYY in one system to YYYY-MM-DD in another to unify sales data analysis.*
    - **Semantic Heterogeneity:** The different meanings or interpretations of the same data across various systems. *Example: Aligning differing patient discharge status terms like "Released" vs. "Discharged" across multiple hospital systems.*
    - **Data Transformation and Extraction:** Modifying data's structure, format, or content to meet the needs of a new application or system, often involving significant changes. *Example: Changing product sizes from "S M L XL" to detailed measurements (e.g., "Small (S) - Chest: 34-36 inches") and converting inventory data into JSON for a modern system.*
    - **Managing Data Integration:** Coordinating the process required to merge data from disparate sources into a unified system. *Example: Integrating customer data from online banking, call centers, and*

branch visits into a single data warehouse, requiring a unified data model because each of these departments uses different systems and standards.

- **Complexity of Integrating Real-Time Data:** Merging and processing data that is continuously generated from various sources, requiring immediate processing and integration for timely decision-making. *Example: Integrating live stock market feeds, transaction data from payment systems, and customer feedback for real-time analytics in a financial app.*
- Other: _____

11. *What solutions have you employed to address data integration challenges?*

    _____

    Which of the following solutions have you employed to address data integration challenges? (Select all that apply.)

    - **Development of Ontologies:** Using tools or guides to create a unified vocabulary for data across systems. *Example: Employing an ontology editor like Protégé to build ontologies that standardise company-wide data terms.*
    - **Cloud-Based Platforms:** Using cloud services for efficient data management and integration. *Example: Utilising AWS for scalable data ingestion and transformation.*
    - **Communication-Centric Approaches:** Applying communication theory to harmonise data meanings across teams. *Example: Holding regular team meetings to ensure consistent data interpretation among stakeholders.*
    - **Automated Continuous Testing:** Employing automated testing frameworks. *Example: Jenkins, Travis CI, or GitHub Actions to regularly check data quality and integrity as part of the development process.*
    - **Agile Workflow Adaptations:** Adjusting development practices to improve data integration outcomes. *Example: Integrating data quality checks into daily stand-ups, adjusting sprint reviews to in-*

clude data integration retrospectives, or adopting pair programming for complex integration tasks.

- Other: _____

Data Collection Challenges and Solutions Questions

Open-ended Question: What data collection challenges have you encountered in your projects? _____

12. Which of the following data collection challenges have you encountered in your projects? (Select all that apply.)

    - **Capturing Diverse Data:** The challenge of collecting structured and unstructured data from varied sources. *Example: Analysing customer sentiment requires collecting purchase data and social media comments.*
    - **Data Collection Method Challenges:** Difficulties arising from using different data gathering tools or techniques. *Example: An agile development team faces challenges with inconsistent data collection from device sensors across various smartphones. Differences in sensor accuracy among smartphones result in gaps in collecting user activity data (what users are doing on their smartphones).*
    - **Informative Data Collection Challenges:** Ensuring that the collected data provides the insights needed for decision-making. *Example: User activity data is collected but lacks details on specific user experience issues.*
    - **Comprehensive Data Collection:** Systematically collecting data across all project aspects. *Example: An e-commerce team might miss out on user experience and satisfaction data while focusing on backend metrics.*
    - **Automation Challenges in Data Collection:** Implementing automated systems for data collection efficiently and accurately. *Example: An automated error logging system incorrectly categorizes user-generated errors, complicating bug prioritisation.*
    - Other: _____

13. *What solutions have you employed to address data collection challenges?*

Which of the following solutions have you employed to address data collection challenges? (Select all that apply.)

- **User-Centered Design Strategies:** Tailoring data collection methods to be intuitive and user-focused. *Example: Designing surveys and interaction trackers based on user behaviour studies.*
- **Automated Continuous Testing and Quality Metrics Dashboards:** Monitoring data quality through automated tools and real-time dashboards. *Example: Implementing continuous testing to quickly spot and rectify data issues.*
- **Automation of Data Collection and Visualisation Toolchains:** Using automated systems for efficient data gathering and visualisation. *Example: Streamlining data workflows to enhance collection accuracy and analysis speed.*
- **Centralised Data Management Systems:** Centralising data handling to improve consistency and data access. *Example: Creating a unified data platform for all stages, from collection to analysis.*
- **Diagnostic Models for Data Collection and Sharing Practices:** Assessing and refining data practices to improve quality and cooperation. *Example: Developing models to optimise data gathering and sharing workflows.*
- **Data-Driven Systems Engineering (DDSE) Methodologies:** Integrating data analysis into system engineering for informed decisions. *Example: Applying DDSE for collaborative and data-informed project development.*
- **Legal Advisor for Data Sensitivity Assessment:** Ensuring data collection complies with data protection laws. *Example: Consulting legal experts to navigate privacy regulations in data practices.*
- **Development of Ontology-Based Approaches:** Enhancing data usefulness through structured data categorization. *Example: Using ontologies to standardise data terms for better interoperability.*

- **Q-Rapids Tool for Valuable Information Acquisition:** Incorporating tools designed for real-time data analysis in development. *Example: Leveraging Q-Rapids for focused data collection in software projects.*
- **Agile Methodology and Data Analyst Collaboration:** Jointly developing data strategies with agile teams and data analysts. *Example: Collaborating on adaptable data collection methods suited to project needs.*
- **Participatory and Co-creative Workshops:** Engaging stakeholders in refining data collection and integration. *Example: Hosting workshops to gather diverse inputs on improving data practices.*
- Other: _____

Data Quality Challenges and Solutions Questions

Open-ended Question: What data quality challenges have you encountered in your projects? _____

14. Which of the following data quality challenges have you encountered in your projects? (Select all that apply.)
    - **Ensuring Data Accuracy and Consistency Across Sources:** Ensuring data from different sources is accurate and presented in a uniform format. *Example: Integrating stock market data reveals inconsistencies in trading volume reports from different exchanges, requiring data normalization for accurate application reporting.*
    - **Completeness of Data:** Ensuring every necessary data field is filled without missing critical information. *Example: User stories for a new feature are found to be missing acceptance criteria during a sprint review, leading to project delays and rework.*
    - **Effective Data Quality Management:** Implementing continuous practices to uphold data integrity and address issues promptly. *Example: Data corruption challenges overwhelm automated correction efforts, making manual reviews necessary and impacting development timelines.*

- **Challenges in Standardising Data Quality Metrics:** Establishing and applying uniform measures for data quality assessment across varied projects. *Example: A team faces difficulty creating data quality metrics that are relevant and applicable to all projects, hindering company-wide quality assessments.*
- Other: _____

15. *What solutions have you employed to address data quality challenges?*

    _____

    Which of the following solutions have you employed to address data quality challenges? (Select all that apply.)

    - **Comprehensive Data Quality Frameworks:** Using established guidelines for ongoing data quality improvement. *Example: Applying Total Data Quality Management (TDQM) principles to ensure data quality across the organisation.*
    - **Automated Data Cleaning and Enrichment Tools:** Utilising software to automatically correct and enhance data. *Example: Deploying tools like Talend or Trifacta for real-time data cleaning and standardisation.*
    - **Data Governance Across Lifecycle:** Creating rules and roles for data management from creation to deletion. *Example: Establishing a governance committee to oversee data security and privacy.*
    - **Standardisation of Data Quality Metrics and Processes:** Setting uniform data quality standards and tracking mechanisms. *Example: Developing a set of data quality KPIs (e.g., accuracy, completeness, consistency, timeliness) for data accuracy and consistency monitored using tools like data quality scorecards.*
    - **Integration and Collection Practices for Quality Assurance:** Embedding quality checks into data collection and integration workflows. *Example: Incorporating validation steps in ETL processes to ensure data integrity from the start.*
    - **Integration of Advanced Analytic Platforms:** Adopting high-tech platforms for enhanced data management and quality. *Example:*

Using Apache Kafka for seamless real-time data integration and quality analysis.

- Other: _____

Data Analysis Challenges and Solutions Questions

Open-ended Question: What data analysis challenges have you encountered in your projects? _____

16. Which of the following data analysis challenges have you encountered in your projects? (Select all that apply.)

    - **Complex Data Sets:** Challenges posed by analysing large or complex datasets. *Example: A team refines vast user data over many sprints to build a machine learning project.*
    - **Real-Time Analysis Requirements:** The challenge of analysing data as it is generated without delay to support immediate decision-making or operational actions. *Example: Developing a real-time user engagement tracker, frequently revisiting their approach to meet analysis speed requirements.*
    - **Analysing Semantic Heterogeneity Data:** Difficulties with data with varying meanings across sources. *Example: A healthcare team struggles to align medical procedure terms across datasets.*
    - **Ensuring Analytical Accuracy:** Keeping data analysis precise and reliable to minimise errors. *Example: A financial project team uses pair programming to boost analysis accuracy, adding unforeseen tasks to their agile process.*
    - **Selection of Appropriate Analytical Techniques:** Choosing suitable analysis methods based on data characteristics. *Example: Debating over using statistical models or machine learning for developing a fraud detection system.*
    - Other: _____

17. *What solutions have you employed to address data analysis challenges?* _____

Which of the following solutions have you employed to address data analysis challenges? (Select all that apply.)

- **Advanced Analytical Tools and Techniques:** Utilising high-level software for in-depth analysis of complex datasets. *Example: Using R and Python for advanced data exploration and Tableau for visualisation.*
- **Real-Time Data Processing Technologies:** Using technologies that enable immediate analysis of live data streams. *Example: Implementing Apache Kafka for streaming and Spark for on-the-fly data processing.*
- **Frameworks and Methods for Semantic Integration:** Applying organised methods to harmonise data meanings across sources. *Example: Utilising RDF (Resource Description Framework) and SPARQL queries with ontologies for consistent data interpretation.*
- **Quality Control Measures in Data Analysis:** Establishing checks to maintain the precision of data analysis outcomes. *Example: Integrating validation steps and cross-validation in analysis workflows to ensure accuracy.*
- **Guidelines for Analytical Technique Selection:** Creating criteria to assist in selecting suitable data analysis methods. *Example: Developing a decision matrix to guide the choice between statistical and machine learning techniques based on data specifics.*
- **Adoption of Machine Learning and AI for Data Analysis:** Utilising AI and machine learning to enhance pattern recognition and prediction in data. *Example: Leveraging AI tools and libraries such as TensorFlow and PyTorch for automated insights and predictive analysis.*
- Other: _____

Recommendations Questions

18. What implications have data management challenges had on your role?

    _____

19. What implications have data management challenges had on your project delivery? _____

20. Please provide any additional comments or insights you may have regarding data management in agile software development. _____

Optional: Stay informed with our research results

To ensure you receive the detailed results of our research, please provide us with your name and email address below. We value your privacy and guarantee that your contact information will only be used to send you the specific results you are interested in.

- Name: _____
- Email Address: _____

Thank you for participating in this survey. Your responses will help us better understand data management challenges and solutions in agile software development.