Bulletin of Business and Economics, 13(2), 974-982
https://bbejournal.com
https://doi.org/10.61506/01.00416

AI-Driven Assistants' Potential for Scaled Agile Software Development

Muhammad Hamza[1], Muhammad Waseem Iqbal[2], Saleem Zubair Ahmad[3]

Abstract

Scaled agile development is commonly used in software engineering to enhance cooperation, productivity, and product quality. Incorporating artificial intelligence (AI) into scaled agile development methods (SADMs) is a promising way to simplify procedures and manage the complexity of software projects. This article examines the impact of AI-powered assistants on the scaled agile framework (SAFe), a popular paradigm for large-scale software development. Our article targets three main objectives: (1) Assessing the obstacles and constraints organizations face while implementing SADMs (2) evaluating the benefits of AI in large-scale situations, and (3) identifying features of SADMs that AI-driven assistants may improve. After conducting a thorough literature analysis, we identified and summarized 18 key difficulties organizations face. Our research identified seven benefits and five barriers to using AI in SADMs. The findings were categorized according to whether they occurred during the development or planning and control stages. We analyzed 15 AI helpers and tools and used them to meet research issues. The findings were categorized according to whether they occurred during the development or planning and control stages. We analyzed 15 AI helpers and tools and used them to meet research issues.

Keywords: SAFe, scaled agile framework, AI, artificial intelligence, tools, assistants, agile, large-scale

1. Introduction

Organizations must react quickly while maintaining agility and growth in today's fast-paced digital world. Implementing proven frameworks for large-scale development, such as SAFe (53% usage, Saklamaeya et al., 2023) and Large-Scale Scrum (6% usage, Saklamaeya et al., 2023), can assist organizations in navigating complex challenges and unknown terrains. These frameworks are essential for organizational success by promoting adaptability, cooperation, and innovation. Implementing transformational aims can provide hurdles, including opposition, procedural complexity, uncertainty, and organizational-wide issues. Many organizations struggle to adopt traditional agile approaches in complex and varied environments. AI can revolutionize several sectors.

This study addresses the constraints of standard-scaled agile development methodologies (SADMs) in managing large-scale projects efficiently. Agile approaches are effective for small-team setups (Saklamaeya et al., 2023) but can struggle with complicated systems with multiple stakeholders, vast code bases, and distant teams. Organizations require creative ways to overcome challenges and optimize development processes. AI has the potential to assist SADMs overcome implementation issues. AI may help organizations enhance efficiency, decision-making, productivity, resource allocation, communication, and data insights. Machine learning (ML) algorithms provide predictive analytics, helping companies anticipate and reduce risks.

We developed the following research questions to guide our study:

RQ1: How might AI assistants successfully help SADMs manage large-scale projects?

RQ2: What are the possible advantages and disadvantages of adding AI into SADMs?

RQ3: What parts of SADMs can AI-powered assistants improve?

The remainder of the paper is organized as follows. We begin with an introduction to SAFe, our selected representation of SADMs, on which we will focus solely. Section 2 provides an overview of relevant studies on large-scale agile development. In Section 3, we provide an overview of our study field and undertake a systematic literature review (SLR) to focus on relevant material. In Section 4, we describe our SLR findings. Sections 4.1 and 4.2 discuss SADM issues and the benefits and drawbacks of incorporating AI inside them. Section 5 categorizes difficulties for various SAFe settings, describes their breadth, and discusses AI-driven assistants that might help resolve them. Section 6 addresses research issues, while Section 8 summarizes major results.

1.1. Introduction to SAFe

SAFe (Jarvinen, 2023) and LeSS (Agia et al., 2023) are development approaches that use agile ideas on a wider scale, including several teams and integrating with existing systems. These techniques prioritize cooperation, change tolerance, adaptable software development, and active client participation, with quick iterations and frequent feedback loops (Omer et al., 2019).

The SAFe framework, version 6.0 (Jarvinen, 2023), combines lean, agile, and DevOps practices to achieve business agility. The focus is on seven core business agility competencies for gaining and maintaining a competitive advantage in a digital world: lean-agile leadership, team and technical agility, agile product delivery, enterprise solution delivery, lean portfolio management, organizational agility, and continuous learning culture (Jarvinen, 2023; Vasilka et al., 2023).

SAFe offers four options for development environments, giving it a flexible strategy for organizations of all sizes and industries:

- **Essential SAFe** is the foundational component of all SAFe systems. This level outlines the necessary SAFe aspects and serves as the foundation for execution. It offers complicated solutions for big teams, at both program and team levels (Jarvinen, 2023).
- **Large-scale SAFe** offers complicated solutions without portfolio management help. It is typically used by organizations with numerous Agile Release Trains (ARTs) collaborating (Jarvinen, 2023).
- **Portfolio SAFe** aims to align agile development with value streams and ARTs. Lean-agile budgeting provides decision-makers with visibility of portfolio and WIP limitations through the Kanban system, as well as objective indicators for management and improvement through bucket-size planning. This level of setup includes portfolios, programs, and teams (Jarvinen, 2023).

[1] Department of Software Engineering, Superior University, Lahore, Pakistan, hamzachh28@gmail.com
[2] Associate Professor, Department of Software Engineering, Superior University, Lahore, Pakistan, waseem.iqbal@superior.edu.pk
[3] Professor Department of Software Engineering, Superior University, Lahore, Pakistan, saleem.zubair@superior.edu.pk

- **Full SAFe** this comprehensive SAFe framework covers all levels, including team, program, large-solution, and portfolio. This comprehensive collection of roles, events, and artifacts is ideal for large organizations that need to synchronize ARTs throughout the company (Jarvinen, 2023).

SAFe is a popular choice in agile development, as seen by the abundance of literature and resources dedicated to it (Vasilka et al., 2023). Our broad knowledge library enables organizations to easily access information and assistance, as well as learn from successful and unsuccessful real-world implementations. In this talk, we will focus on SAFe as an example of SADMs and analyze its principles and benefits with AI.

2. Literature Review

Our research on problems in SADMs led us to several articles that addressed similar issues in different situations (refer to Section 4.1). Some articles (Omer et al., 2018; Sinha et al., 2020; Paolo et al., 2022) used systematic literature reviews (SLRs) to identify problems in SADMs, whereas others (Conboy et al., 2019) relied on structured interviews and empirical data. Studies have identified unique issues, such as stakeholder management and recurring concerns and trends (Fucci et al., 2018).

While researching the benefits of using AI in SADMs, we came across several papers. Several papers (Xin et al., 2019) have presented or reported experiences with pair programming. We found a publication (Jarrahi et al., 2018) that did an SLR to gain a comprehensive understanding of the research domain and identified prospective areas for AI-driven assistants to provide support.

The study found 79 issues associated with large-scale agile development, including 41 new challenges and 38 enhanced ones. The planned future effort (Omer et al., 2019) entails using a large-scale agile pattern language to handle recurring difficulties without identifying specific tools. Sinha et al. (2020) classified 11 obstacles as internal and external elements. The authors utilized their findings as a foundation for future work, focusing on overcoming hurdles and leveraging success characteristics (Paolo et al., 2022). Undertook a comprehensive literature analysis and empirical research, but cautioned that their findings may not be generalizable due to the small sample size and observational nature.

Although they did not provide particular remedies, they did make recommendations for further research (Conboy et al., 2019). Identified nine obstacles across 13 situations and provided ideas for resolution. They acknowledged the complexity of total removal owing to several causes but did not include AI-driven assistant solutions (Kasauli et al., 2021). Conducted qualitative interviews to identify 24 issues and provide solutions based on SAFe, LeSS, and case firms. They also identified gaps in the literature and suggested potential remedies. The goal is to describe the motivations, advantages, and problems of implementing SADMs, without proposing particular solutions and emphasizing the necessity for future quality assessments.

These sources had a considerable impact on our study, leading to a synthesis of information on obstacles in large-scale development. We propose AI-driven helpers to address these difficulties. During the SLR process, no articles addressed issues in large-scale development using AI-driven assistants. Section 7 outlines the limits of our research, which may explain this absence.

3. Research Method

This article evaluates the current state of AI-driven assistants inside the SAFe framework. We seek to showcase current research, developments, and uses of AI in supporting agile approaches in large-scale software development settings. We analyze the merits and shortcomings of AI-driven assistants to identify areas for improvement and future research.

A systematic literature review (SLR) was conducted to summarize relevant. We followed the systematic review methods proposed by (Elbasheer et al., 2022). We took a rigorous and extensive approach to our investigation. We shall address the research questions in Section 1 utilizing the findings collected through this strategy. With the parameters established, we began the SLR procedure. Figure 1 depicts the six steps of a literature review undertaken in mid-2023.

- **Initial search**

In the first phase of the SLR process, we collected 2159 objects. We searched Science Direct, Springer Link, and ACM Digital Library for publications in Computer Science, Informatics, and Software Engineering, filtering them based on defined inclusion criteria.

- **Duplicate elimination.**

Duplicate elimination involved normalizing all findings into a single literature list and eliminating duplicates. We removed 313 publications that appeared several times, resulting in a literature list of 1846 items.

- **Screening is done based on titles and keywords**

We decreased the literature list from 1846 to 112 pieces during this step. The increased interest in AI research has resulted in increased publications and case studies from numerous areas, including medicine and construction. We focused our literature selection on SADMs and their implementation in Computer Science and Information Systems.

- **Abstract-based screening**

Additional papers were analyzed based on their abstracts. We decreased our reading list from 112 to 35 articles.

- **Content-based screening**

During this stage, we thoroughly reviewed the selected articles for their substance. We analyzed 35 publications and excluded those that did not fit our research topic. At this point, 24 articles remained on the literature list.

- **Snowballing**

We investigated the selected publications' references to find other sources. We ensured that we included sources that fulfilled our research requirements and were not available in the selected databases. At this step, we added six items, resulting in thirty primary sources (see Section 4).

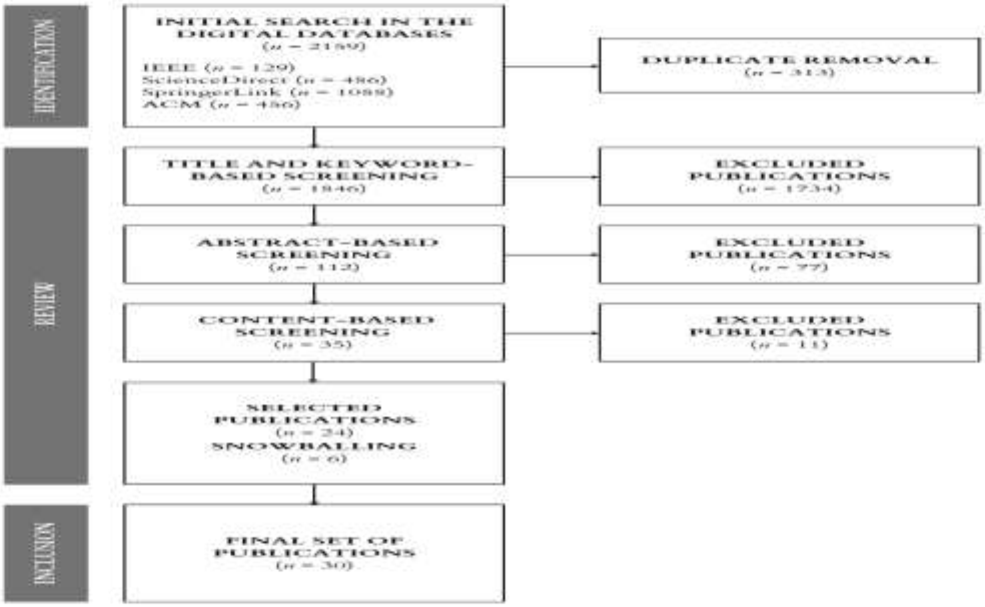

Figure 1: Systematic literature review process

4. Results

4.1. Challenges of Scaled Agile Development

SADMs are becoming more widespread in software development organizations (Saklamaeva et al., 2023). While scaling agile development offers potential benefits, it can also present obstacles such as collaboration, communication, and adaptability (Conboy et al., 2019). SADMs like SAFe offer proven workflow patterns and expandable tools, making them popular solutions for overcoming these challenges. Empirical research on the acceptance, usage, success, and challenges of these strategies is still in its early stages (Omer et al., 2019; Omer et al., 2018; Brühl et al., 2022). Analyzed 13 big projects from multinational organizations over 15 years. The study found that successful SADM implementation requires more than just adhering to the framework's requirements. Some firms and organizations found success using a certain approach, while others improved their performance and efficiency by switching to a different method. Others completely abandoned the usage of SADMs due to various causes. Figure 2 shows a visual representation of SADM implementation over time (Conboy et al., 2019).

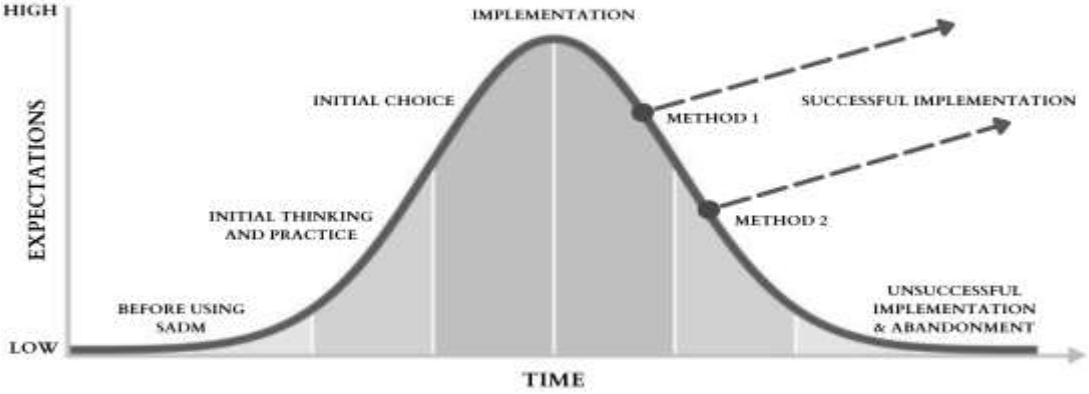
HIGH
EXPECTATIONS
LOW
TIME
IMPLEMENTATION
INITIAL CHOICE
METHOD 1
SUCCESSFUL IMPLEMENTATION
METHOD 2
INITIAL THINKING AND PRACTICE
BEFORE USING SADM
UNSUCCESSFUL IMPLEMENTATION & ABANDONMENT

Figure 2: The use of large-scale agile frameworks

15-year research (Conboy et al., 2019; Kasauli et al., 2021), and a proposal for an AI-driven solution (Ameta et al., 2022) identify recurring obstacles in organizations using SADMs (Omer et al., 2018). They performed an SLR to identify stakeholders in SADMs. The assessment identified 79 issues, divided into 11 categories.

A poll of software practitioners examined the reasons for adopting SADMs, possible advantages, and satisfaction with their deployment in scaled contexts (Omer et al., 2019). A focused on documenting recurring issues and trends in large-scale development. The authors developed a pattern language and conducted structured interviews with 14 large-scale agile development specialists from 10 organizations to gain insights into real-world difficulties and concerns (Putta et al., 2021).

An SLR and SWOT analysis were conducted to assess the impact of SADM adoption. In addition, (Paolo et al., 2022) conducted an SLR and empirical research to acquire a deeper knowledge of adopting SADMs, with a focus on SAFe. Input data from 25 respondents from 17 organizations in eight countries revealed issues in decision-making, organizational frameworks, and technical and management capabilities. The case studies on three firms across sectors. These studies used structured interviews to assess existing concerns, needs, and implementation obstacles (Fucci et al., 2018).

We summarize the issues observed in large-scale development settings using the SLR. When considering the issues, it's important to notice that we don't address any major ones openly. The issues differ in their nature, categorization, and significance to certain fields. Our list of obstacles faced by organizations using SADMs is not exhaustive, but we strive to accurately reflect real-world situations while acknowledging the limits indicated in Section 3. In Section 5, we analyze the highlighted difficulties and use AI to propose ideas for solutions.

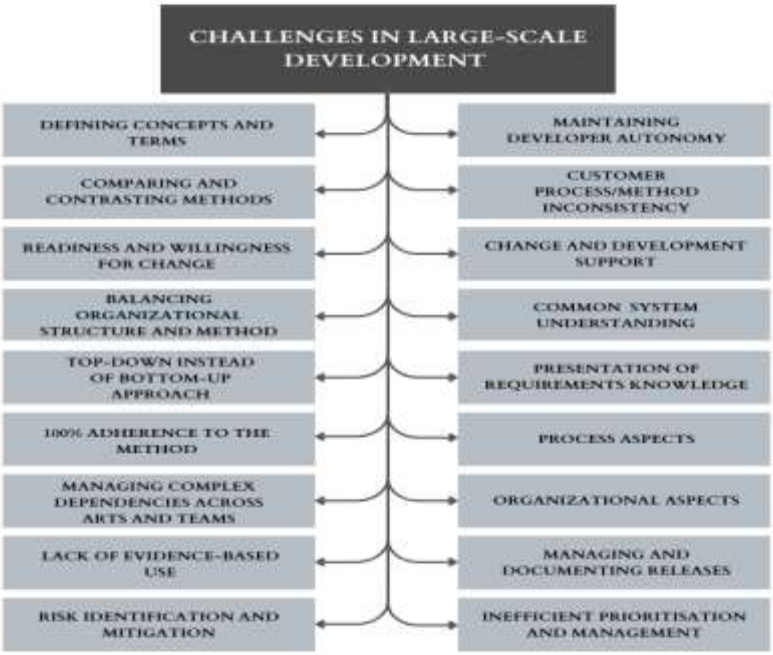
CHALLENGES IN LARGE-SCALE DEVELOPMENT
DEFINING CONCEPTS AND TERMS
MAINTAINING DEVELOPER AUTONOMY
COMPARING AND CONTRASTING METHODS
CUSTOMER PROCESS/METHOD INCONSISTENCY
READINESS AND WILLINGNESS FOR CHANGE
CHANGE AND DEVELOPMENT SUPPORT
BALANCING ORGANIZATIONAL STRUCTURE AND METHOD
COMMON SYSTEM UNDERSTANDING
TOP-DOWN INSTEAD OF BOTTOM-UP APPROACH
PRESENTATION OF REQUIREMENTS KNOWLEDGE
100% ADHERENCE TO THE METHOD
PROCESS ASPECTS
MANAGING COMPLEX DEPENDENCIES ACROSS ARTS AND TEAMS
ORGANIZATIONAL ASPECTS
LACK OF EVIDENCE-BASED USE
MANAGING AND DOCUMENTING RELEASES
RISK IDENTIFICATION AND MITIGATION
INEFFICIENT PRIORITISATION AND MANAGEMENT

Figure 3 identified issues during the SADM implementation

4.2. The Pros and Cons of Using AI in SADM

AI in software development improves efficiency, quality, and innovation at all phases (Xin et al., 2019; Jarrahi et al., 2018). According to (Brühl, 2022), disruptive technologies like AI, ML, and Blockchain are driving shorter product life cycles, fragmented value chains, and new organizational structures such as value creation networks, platform solutions, and cluster organizations. AI-powered assistants may automate repetitive processes like code review, testing, and error detection, resulting in considerable labor savings and shorter development cycles (Xin et al., 2019). AI can recognize patterns and trends in data, allowing developers to make educated decisions and predict possible concerns. AI-driven algorithms may improve code performance, recommend improvements, generate documentation, expedite communication, and foster knowledge sharing within development teams.

AI may automate software development through pair programming, involving a developer and an AI helper. Incorporating AI in development has significant benefits, but also presents obstacles. To overcome this, they propose automating cooperation between humans and AI. Developers would continue to follow conventional procedures, with the AI-driven assistant working in the background to give help when challenges arise. This combination aims to enhance productivity and quality by minimizing repetitive operations and guiding new developers to think and operate like professionals (Jarrahi et al., 2018). We investigated the use of AI helpers in planning and control domains. An SLR revealed three significant areas where AI helpers might be beneficial.

The author concluded that using AI assistants to aid people will result in considerable commercial success (Mikalef et al., 2021). The study examined the capabilities of AI helpers in large-scale deployments. The study identified three types of capabilities: tangible (data and technologies), human (technical and business skills), and intangible (coordination and adaptability). This provides insight into the broader implications of implementing these technologies. The study identifies frequent hurdles in integrating AI, offers guidelines for moving to its usage, and highlights possible benefits for organizations. Figures 4 and 5 summarize the advantages and challenges of applying AI.

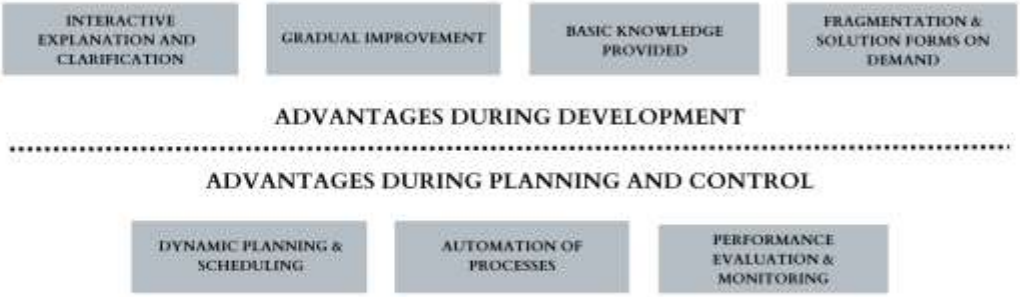
INTERACTIVE EXPLANATION AND CLARIFICATION
GRADUAL IMPROVEMENT
BASIC KNOWLEDGE PROVIDED
FRAGMENTATION & SOLUTION FORMS ON DEMAND
ADVANTAGES DURING DEVELOPMENT
ADVANTAGES DURING PLANNING AND CONTROL
DYNAMIC PLANNING & SCHEDULING
AUTOMATION OF PROCESSES
PERFORMANCE EVALUATION & MONITORING

Figure 4: Identified the benefits of using AI during SADM implementation

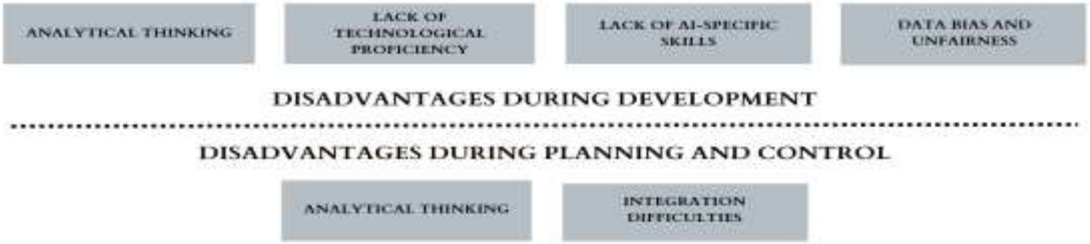
ANALYTICAL THINKING
LACK OF TECHNOLOGICAL PROFICIENCY
LACK OF AI-SPECIFIC SKILLS
DATA BIAS AND UNFAIRNESS
DISADVANTAGES DURING DEVELOPMENT
DISADVANTAGES DURING PLANNING AND CONTROL
ANALYTICAL THINKING
INTEGRATION DIFFICULTIES

Figure 5: Identified the downsides of applying AI during SADM implementation

5. Extending evaluation results to SAFe environments

This study aims to link highlighted difficulties to SAFe's structural organization at various levels, as indicated in the introduction. We aim to identify the specific spots inside SAFe where these impediments are most likely to arise. SAFe challenges are compiled at each level, allowing for a comprehensive investigation of potential issues. This is justified as the complete level of the SAFe hierarchy includes all subordinate elements. The sections below provide a detailed overview of the difficulties at hand. The research will provide insights and recommendations for AI-driven assistants to solve these difficulties.

Table 1: Identified problems in SAFe

Essential (E)	Large Solution (LS)	Portfolio (P)
• Define concepts and terminology. • Adaptability and readiness to change. • Ensuring developer autonomy. • Inconsistency between customer processes and methods. • Presenting requirements knowledge • Discussing process aspects. • Risk Assessment and Mitigation.	• Define concepts and terminology. • Risk Assessment and Mitigation. • Change development and support. • Universal System Understanding • Organizing and recording releases • Handling Complex Depending between ARTs and Team	• Define concepts and terminology. • Risk Assessment and Mitigation. • Comparing and comparing methods. • Balancing organizational structure and methods. • Top-down vs bottom-up approach. • 100% adherence to the method. • Lack of evidence-based use. • Organizational aspects. • Inefficient prioritization and management

5.1. Difficulties at an essential level

- **Defining concepts and terms**

Papers introducing approaches such as SAFe and LeSS (Ramadhina et al., 2023) provide clear definitions of their basics. However, using these strategies in diverse circumstances might lead to unclear implementation recommendations (Conboy et al., 2019). Companies sometimes have misconceptions regarding the principles and procedures of these methodologies, and there are significant variances in their interpretation and use across different industries. The use of abstract terms has been a recurring concern in our study. Transitioning to SADMs might be challenging due to a lack of thorough explanations.

- **To successfully transition to agile methods**

The organizations and employees must be open to change. While employees may embrace changing software processes, they may not be ready to adopt specific methods. Frameworks such as SAFe and LeSS (Ramadhina et al., 2023) offer structures and processes but lack guidance on assessing overall readiness for agile transformation at scale (Conboy et al., 2019). The authors of (Sinha et al., 2020) found considerable skepticism. To successfully transition to agile methods, organizations and employees must be open to change. While employees may embrace changing software processes, they may not be ready to adopt specific methods. Frameworks such as SAFe and LeSS (Ramadhina et al., 2023) offer structures and processes but lack guidance on assessing overall readiness for agile transformation at scale (Conboy et al., 2019). The authors of (Sinha et al., 2020) found considerable skepticism.

- **Maintaining developer autonomy**

In large contexts is becoming increasingly challenging. SADMs exacerbate the problem by adding limitations and inflexibility. The authors of (Conboy et al., 2019) report that developers' recommendations for tool and process enhancements have been rejected owing to apparent conflict with the new method's implementation. According to (Sinha et al., 2020), cultural and language challenges might arise when development teams are distributed throughout many nations.

- **Customer process/method discrepancy**

Implementing a new SADM might be challenging as it requires transforming the organization's preset procedures and structures (Conboy et al., 2019). The organizations may need to create collaboration agreements with clients to generate software inside a specified development framework. Even with tight client ties, there may still be a significant distance between customers and developers (Conboy et al., 2019). Development teams may fail to understand the customer's perspective and communicate how their work directly helps them, leading to a gap. Crafting user stories that benefit the consumer might be tough. Completing and demonstrating intricate stories within a single sprint or iteration can be challenging, as can maintaining reusable customer insights within complex product families. Changes may require repetitive work to obtain similar customer-related information.

- **Presentation of requirement knowledge**

In this context, obstacles, as described in (Kasauli et al., 2021), include concerns such as managing people in hierarchical tiers rather than an organizational decomposition, as well as establishing acceptable need thresholds. Furthermore, we drew knowledge from (Fucci et al., 2018), where the authors subjected challenges such as dealing with an overload of data (dealing with the gathering, searching, and evaluation of vast amounts of information), coping with the constraints of the chosen development assistants, and handling dependencies between requirements.

- **System Aspects**

This challenge focuses on identifying prevalent challenges in SADM development processes. The challenges include prioritizing high-priority tasks (Kasauli et al., 2021), determining the completeness of requirements (Kasauli et al., 2021), managing requirements using various tools and levels of detail (Kasauli et al., 2021), and establishing a clear quality threshold for release readiness (Kasauli et al., 2021). The development (Sinha et al., 2020) teams struggle with collaboration, excessive commitment, project abandonment, and motivation. Additionally, there is a lack of interpersonal communication, agile coaching, and linguistic difficulties (Ameta et al., 2022).

- **Risk Assessment and Mitigation**

This category of challenges focuses on identifying and assessing risks that can impact a project's timeline, quality, and finances (Omer et al., 2018). Predicting future risks is challenging due to the inherent unpredictability, temporal relationships, and dynamic nature of software (Kitchenham et al., 2021).

5.2. Difficulties at the Large-Solution Level

- **Helping with change and development**

i. Managing experimental or poorly specified needs.
ii. Synchronizing development efforts among teams in large-scale contexts can be complicated, limiting agility and speed.
iii. Requirements stated at the start of a sprint may become old and no longer fit the solution.

Highlight the management of unresolved dependencies. With so many cross-functional teams dependent on data and information, unresolved connections between data points can stymie development (Fountaine et al., 2019) Identified many challenges when an organization attempts to transition from inflexible and risk-averse to agile, experimental, and adaptive (Kitchenham et al., 2012). Emphasized how software engineering's major focus remains on solving issues or providing new functionality, rather than adapting to ever-changing conditions.

- **Universal system understanding**

Large-scale agile development environments can lack a common knowledge of the system.

Challenges in this setting include insufficient documentation for testing and storytelling, misunderstanding between system and component levels, and insufficient monitoring and maintenance.

- **Controlling and documenting releases**

This challenge tackles both technological and organizational challenges, including collaboration and communication issues across ARTs, development teams, and stakeholders (Omer et al., 2018). Maintaining a comprehensive record of product or solution changes is essential as it evolves. Failure to do so may result in release misalignment or delays providing lightweight documentation that meets all needs is a recurring difficulty.

- **Managing complicated connections across ARTs and teams**

As organizations increase their solutions and procedures, the interdependence across ARTs and teams typically grows as well. This problem is commonly depicted as a chain of codependent tasks, with one team's success relying on another. Inadequate project management may lead to delays, reduced productivity, and even project failure (Omer et al., 2018).

5.3. Portfolio-level challenges

- **Comparing and contrasting approaches**

Many companies struggle to choose the right SADM due to a lack of a comparative evaluation model. According to Kieran et al. (2019), the decision-making process can be ad hoc and unclear (Paolo et al., 2022). Additionally, many respondents found the implementation of SADMs to be complex and difficult to understand.

- **Balancing organizational structure and methodology**

Systems, procedures, and proprietary tools. These frameworks are altering continually in response to external competition and regulatory demands, making a one-size-fits-all strategy problematic (Sinha et al., 2020) noted a lack of support and commitment from the senior management

- **Top-down rather than bottom-up strategy**

Many implementations have followed either a bottom-up or top-down strategy, rather than a hybrid of the two (Sinha et al., 2019) top-down techniques have produced mixed outcomes, emphasizing many issues related to the implementation of SADMs. One key difficulty is that their execution frequently leads to top-down organizational control, resulting in a structure similar to a waterfall technique that lacks genuine agility and adaptability.

- **100% devotion to the technique**

When a formal technique such as SAFe is adopted, it is usual to measure the success of agile transformation based on how closely the organization conforms to the approach rather than the value it provides. It has been noted that SADMs typically meet difficulties or challenges during the final 20-30% of development operations, with the last 5% accounting for an extremely significant percentage of the work and stress (approximately 80-90%) (Conboy et al., 2019). Furthermore, a significant concern raised in 30% of the selected articles in (Sinha et al., 2020) is a lack of suitable agile training for scaling development environments.

- **Lack of evidence-based usage**

There are few empirical case studies investigating the actual implementation of prescriptive concepts in SADMs. They experienced circumstances when employees faced considerable obstacles, and they struggled to discover important information regarding their solution inside the documentation of the approach they were utilizing.

- **Organizational aspects**

There are three common issues with organized scaled environments:

1. Disparity between plan-oriented, document-heavy systems and value-driven, agile teams.
2. Requirement-based validation and verification techniques are incomplete and gradual.
3. Prioritizing infrastructure enhancements for timely success.

Many scaled development environments also faced the issue of unequal job distribution. Because management gives responsibilities to teams, they may be allocated unevenly among team members. In (Carleton et al., 2020; Zhan et al., 2021; Zimmermann et al., 2020; Song et al., 2022), we encountered the long-standing issue of "quick wins". They highlight the reality that organizations should focus on establishing a portfolio of projects over a longer time to maximize their return on investment.

- **Inefficient prioritization and management**

This difficulty involves challenges such as difficulties in identifying clear and visible priorities that are consistent with the organization's strategic goals, as well as dealing with a loss of managerial control (Omer et al., 2018). This might be due to a lack of automated support for effort estimates (Kitchenham, 2012). Failure to handle these challenges properly can result in poor resource allocation, leading to delays, misalignments, increased expenditures, and project cancellation (Fucci et al., 2018).

6. Discussion

This article examines the integration of AI helpers in SADMs. Our study approach, SLR, yielded noteworthy discoveries and insights, which we highlight. Our research focuses largely. This study examines the possible benefits and risks of incorporating AI-driven assistants into SAFe, a software development methodology.

We explored how AI helpers may help SADMs manage large-scale projects successfully. Second, we explored the possible benefits of integrating AI in SADMs. Finally, we investigated how AI-powered assistants may improve some elements of SADMs.

RQ1: How might AI assistants successfully help SADMs manage large-scale projects?

AI assistants can help manage large-scale projects in SADMs.

Section 4.1 presents a comprehensive list of significant problems discovered in the literature. Balancing organizational structure and approach, retaining developer autonomy, and achieving shared system understanding are some of the problems. However, integrating AI-driven assistants can positively address these difficulties. Given the scalability of SAFe to varied organizational sizes, our initial purpose was to categorize the identified challenges according to the SAFe tiers (essential, large-solution, and portfolio) where they are most likely to arise.

Our SLR disclosed a multiplicity of articles (citation rates specified in Figure 6) that either tackled these difficulties explicitly through the deployment of specialist AI-driven assistants or gave broad counsel on how businesses should manage these challenges effectively by incorporating AI. In Section 5, we illustrate a relationship between the listed difficulties, their potential solutions including AI-driven assistants, and recommendations for the adoption of AI. This research covers all identified difficulties at each SAFe level, including the highest and most thorough level.

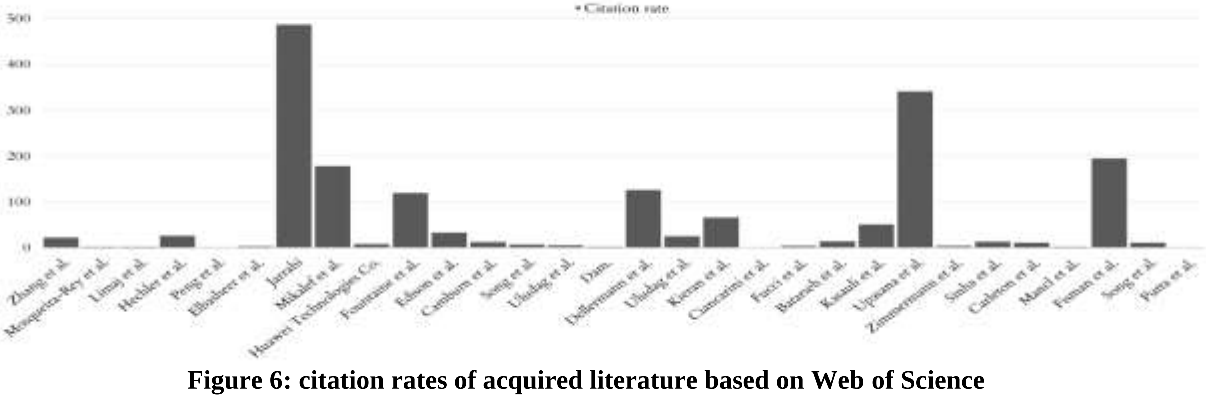

Figure 6: citation rates of acquired literature based on Web of Science

RQ2: What are the possible benefits and difficulties of implementing AI into SADMs?

We focused on identifying the benefits and limitations of integrating AI into SADMs. Our study findings, based on limited literature obtained during the SLR process, are described in Section 4.2. Our research revealed seven key benefits, including process automation, dynamic planning, scheduling, and iterative enhancement. We identified six problems related to critical thinking, biases, and injustice in data, and integration. Our study and practical investigations show that integrating AI into SADMs can improve organizational performance and potential.

RQ3: What parts of SADMs can AI-powered assistants improve?

After analyzing the SLR findings in Section 4, we categorized the identified assistants to help them. Organizations and businesses are addressing particular concerns. This part categorizes these helpers by Usage Domains, which include human support, risk prediction, issue resolution, and more. Upon additional analysis, we discovered that several helpers overlap many areas within the category. AI helpers are very adaptable and capable of addressing many issues.

In Section 5, we give a classification of common difficulties among SAFe setups. This technique helps organizations identify difficulties particular to the SAFe configuration they have installed. Our classification helps organizations strategically implement AI-driven technologies and successfully handle diverse issues throughout their systems.

7. Limitations and Threats to Validity

During the SLR process, we focused on SAFe, the most generally deployed framework. Our research focused on actual experiences, success and failure stories related to SAFe, and the use of AI-driven assistants in various organizational areas. The study focuses on SAFe as the major SADM, however understanding other agile frameworks might widen its usefulness and ramifications.

Sections 5.1-5.3 describe AI-driven assistants and recommendations as potential solutions, however, they may not solve all of the difficulties highlighted. The scope of these issues makes it difficult to determine whether they are effective as absolute solutions. We claim to have a relatively accurate estimate of their potential based on literature and assertions. Evaluating their usefulness in real-world scenarios is currently unfeasible. Our goal was to explore effective solutions for emerging difficulties using cutting-edge technology. Another drawback of our study is the lack of a hierarchical categorization or hierarchy of detected attributes in AI-driven assistants. Our research focused on AI's ability to handle issues in SADMs, rather than comprehensively analyzing or categorizing these traits. Our analysis did not provide a thorough taxonomy of the found characteristics.

Our investigation and literature choices were done in mid-2023. Recent advancements may have an impact on our study findings, which we have not taken into consideration. Although we have invested significantly. Despite efforts to generate relevant search keywords and execute a systematic database search, not all relevant publications may have been discovered.

A reverse search of analyzed publications revealed more material. Despite our best efforts, crucial papers may nevertheless be overlooked.

8. Conclusions and Future Work

This article provides an overview of AI-driven assistants in SADMs, namely SAFe. It categorizes them based on application domains and identifies difficulties they can address. This study explains frequent issues faced by SADMs, focusing on SAFe specifically. Major obstacles in large-scale software development include collaboration, resource utilization, and opposition to change.

We present AI-driven solutions that partially solve the concerns identified in the paper's introduction. To achieve this goal, we conducted an initial assessment of the benefits and drawbacks of incorporating AI into SADMs. Our data indicate that the benefits outweigh the drawbacks. However, organizations and companies must be cautious and knowledgeable when implementing AI in their operational frameworks. We've compiled a list of AI helpers that can address some of the difficulties described above. Our evaluation of these helpers highlights their potential to improve several elements of SAFe. Furthermore, we have not just emphasized. We have evaluated the technological possibilities of AI-driven assistants, as well as the necessary organizational components for effective deployment. Developed recommendations based on comprehensive research and real-world examples, emphasizing proven best practices.

The outcomes of our study offer various directions for further investigation. Future research should aim to develop AI helpers that can improve many aspects of SAFe, such as portfolio management. What technologies may enhance Design Thinking and Lean UX for Agile Product Delivery? Empirical research in real-world SADM contexts can provide significant insights into the practical application and obstacles of adopting AI helpers. Integrating future technologies, such as ML models or NLP, to improve the capabilities of these assistants is a viable path forward.

AI helpers might revolutionize large-scale project design and execution. Empirical studies indicate that adopting AI-driven solutions can result in improved communication, decision-making, and project outcomes are key for firms to stay innovative and competitive. This study suggests that successfully integrating AI helpers requires a comprehensive approach that takes into account both benefits and downsides.

References

Ameta, U., Patel, M., & Sharma, A. K. (2022, October). Scaled Agile Framework Implementation in Organizations', its Shortcomings and an AI Based Solution to Track Team's Performance. In *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)* (pp. 1-7). IEEE.

Batarseh, F. A., & Gonzalez, A. J. (2018). Predicting failures in agile software development through data analytics. *Software Quality Journal*, *26*, 49-66.

Brühl, V. (2022). Agile methods in the German banking sector: some evidence on expectations, experiences and success factors. *Journal of business economics*, *92*(8), 1337-1372.

Carleton, A. D., Harper, E., Menzies, T., Xie, T., Eldh, S., & Lyu, M. R. (2020). The AI Effect: Working at the Intersection of AI and SE. *IEEE Software*, *37*(4), 26-35.

Ciancarini, P., Kruglov, A., Pedrycz, W., Salikhov, D., & Succi, G. (2022, May). Issues in the adoption of the scaled agile framework. In *Proceedings of the 44th international conference on software engineering: software engineering in practice* (pp. 175-184).

Conboy, K., & Carroll, N. (2019). Implementing large-scale agile frameworks: challenges and recommendations. *IEEE software*, *36*(2), 44-50.

Dam, H. K. (2019). Empowering software engineering with artificial intelligence. In *Service Research and Innovation: 7th Australian Symposium, ASSRI 2018, Sydney, NSW, Australia, September 6, 2018, and Wollongong, NSW, Australia, December 14, 2018, Revised Selected Papers 7* (pp. 22-32). Springer International Publishing.

Edison, H., Wang, X., & Conboy, K. (2021). Comparing methods for large-scale agile software development: A systematic literature review. *IEEE Transactions on Software Engineering*, *48*(8), 2709-2731.

Elbasheer, M., Longo, F., Nicoletti, L., Padovano, A., Solina, V., & Vetrano, M. (2022). Applications of ML/AI for decision-intensive tasks in production planning and control. *Procedia Computer Science*, *200*, 1903-1912.

Fisman, D., & Rosu, G. (2022). *Tools and Algorithms for the Construction and Analysis of Systems: 28th International Conference, TACAS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2–7, 2022, Proceedings, Part I* (p. 583). Springer Nature.

Fountaine, T., McCarthy, B., & Saleh, T. (2019). Building the AI-powered organization. *Harvard Business Review*, *97*(4), 62-73.

Fucci, D., Palomares, C., Franch, X., Costal, D., Raatikainen, M., Stettinger, M., ... & Maalej, W. (2018, October). Needs and challenges for a platform to support large-scale requirements engineering: A multiple-case study. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (pp. 1-10).

Jarrahi, M. H. (2018). Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making. *Business horizons*, *61*(4), 577-586.

Järvinen, M. (2023). The Benefits and Challenges of Scaled Agile Framework in the IT Industry-case study: company x.

Kasauli, R., Knauss, E., Horkoff, J., Liebel, G., & de Oliveira Neto, F. G. (2021). Requirements engineering challenges and practices in large-scale agile system development. *Journal of Systems and Software*, *172*, 110851.

Kitchenham, B. A. (2012, September). Systematic review in software engineering: where we are and where we should be going. In *Proceedings of the 2nd international workshop on Evidential assessment of software technologies* (pp. 1-2).

Limaj, E., & Bernroider, E. W. (2022). A taxonomy of scaling agility. *The journal of strategic information systems*, *31*(3), 101721.

Lu, M., & Qiu, J. L. (2022). Empowerment or warfare? dark skin, AI camera, and Transsion's patent narratives. *Information, Communication & Society*, *25*(6), 768-784.

Mancl, D., & Fraser, S. D. (2021, June). The Future of Software Engineering: Where Will Machine Learning, Agile, and Virtualization Take Us Next?. In *International Conference on Agile Software Development* (pp. 222-230). Cham: Springer International Publishing.

Mikalef, P., & Gupta, M. (2021). Artificial intelligence capability: Conceptualization, measurement calibration, and empirical study on its impact on organizational creativity and firm performance. *Information & management*, *58*(3), 103434.

Mosqueira-Rey, E., Pereira, E. H., Alonso-Ríos, D., & Bobes-Bascarán, J. (2022, April). A classification and review of tools for developing and interacting with machine learning systems. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing* (pp. 1092-1101).

Peng, X., Xing, Z., & Sun, J. (2019). AI-boosted software automation: learning from human pair programmers. *Science China. Information Sciences*, *62*(10), 200104.

Ploennigs, J., & Berger, M. (2023). AI art in architecture. *AI in Civil Engineering*, *2*(1), 8.

Putta, A., Uludağ, Ö., Hong, S. L., Paasivaara, M., & Lassenius, C. (2021, October). Why do organizations adopt agile scaling frameworks? a survey of practitioners. In *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (pp. 1-12).

Ramadhina, F. A., & Raharjo, T. (2023). Scaled Agile Method Design: Case Study of a SaaS Company in Indonesia. *J-Icon: Jurnal Komputer dan Informatika*, *11*(2), 165-173.

Saklamaeva, V., & Pavlič, L. (2023). The potential of ai-driven assistants in scaled agile software development. *Applied Sciences*, *14*(1), 319.

Sinha, R., Shameem, M., & Kumar, C. (2020, February). SWOT: Strength, weaknesses, opportunities, and threats for scaling agile methods in global software development. In *Proceedings of the 13th Innovations in Software Engineering Conference (formerly known as India Software Engineering Conference)* (pp. 1-10).

Song, B., Gyory, J. T., Zhang, G., Zurita, N. F. S., Stump, G., Martin, J., ... & Cagan, J. (2022). Decoding the agility of artificial intelligence-assisted human design teams. *Design Studies*, *79*, 101094.

Song, B., Zurita, N. S., Zhang, G., Stump, G., Balon, C., Miller, S. W., ... & McComb, C. (2020, May). Toward hybrid teams: A platform to understand human-computer collaboration during the design of complex engineered systems. In *Proceedings of the design society: DESIGN conference* (Vol. 1, pp. 1551-1560). Cambridge University Press.

Uludağ, Ö., Harders, N. M., & Matthes, F. (2019, July). Documenting recurring concerns and patterns in large-scale agile development. In *Proceedings of the 24th European Conference on Pattern Languages of Programs* (pp. 1-17).

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

Zhang, G., Raina, A., Cagan, J., & McComb, C. (2021). A cautionary tale about the impact of AI on human design teams. *Design Studies*, *72*, 100990.

Zimmermann, A., Schmidt, R., & Sandkuhl, K. (2020). Strategic challenges for platform-based intelligent assistants. *Procedia computer science*, *176*, 966-975.