applied sciences

Article

The Potential of AI-Driven Assistants in Scaled Agile Software Development

Vasilka Saklamaeva * and Luka Pavlič

Faculty of Electrical Engineering and Computer Science, University of Maribor, 2000 Maribor, Slovenia; luka.pavlic@um.si
* Correspondence: vasilka.saklamaeva@um.si

Abstract: Scaled agile development approaches are now used widely in modern software engineering, allowing businesses to improve teamwork, productivity, and product quality. The incorporation of artificial intelligence (AI) into scaled agile development methods (SADMs) has emerged as a potential strategy in response to the ongoing demand for simplified procedures and the increasing complexity of software projects. This paper explores the intersection of AI-driven assistants within the context of the scaled agile framework (SAFe) for large-scale software development, as it stands out as the most widely adopted framework. Our paper pursues three principal objectives: (1) an evaluation of the challenges and impediments encountered by organizations during the implementation of SADMs, (2) an assessment of the potential advantages stemming from the incorporation of AI in large-scale contexts, and (3) the compilation of aspects of SADMs that AI-driven assistants enhance. Through a comprehensive systematic literature review, we identified and described 18 distinct challenges that organizations confront. In the course of our research, we pinpointed seven benefits and five challenges associated with the implementation of AI in SADMs. These findings were systematically categorized based on their occurrence either within the development phase or the phases encompassing planning and control. Furthermore, we compiled a list of 15 different AI-driven assistants and tools, subjecting them to a more detailed examination, and employing them to address the challenges we uncovered during our research. One of the key takeaways from this paper is the exceptional versatility and effectiveness of AI-driven assistants, demonstrating their capability to tackle a broader spectrum of problems. In conclusion, this paper not only sheds light on the transformative potential of AI, but also provides invaluable insights for organizations aiming to enhance their agility and management capabilities.

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

Citation: Saklamaeva, V.; Pavlič, L. The Potential of AI-Driven Assistants in Scaled Agile Software Development. *Appl. Sci.* **2024**, *14*, 319. https://doi.org/10.3390/app14010319

Academic Editor: Christos Bouras

Received: 8 November 2023
Revised: 14 December 2023
Accepted: 22 December 2023
Published: 29 December 2023

1. Introduction

Since today's digital landscape is becoming more fast-paced and evolving by the minute, organizations face the challenge of adapting to change while still preserving their agility and growing size. Implementing well-known and verified frameworks to support large-scale development, such as the scaled agile framework (SAFe) (reportedly holding the largest usage share, at 53%, according to ref. [1]) and Large-Scale Scrum (LeSS) (reporting a 6% usage share as per ref. [1]), can help these organizations navigate complex challenges and unknown terrains. Frameworks like these serve as foundational building blocks for achieving the agility, cooperation, and innovation required for organizational success. Their implementation to achieve the organization's transformative goals is often followed by an increasing number of challenges, ranging from resistance and process complexities to overall confusion and organization-wide difficulties. It is estimated that a significant majority of enterprises encounter notable difficulties when attempting to apply conventional agile methodologies in such expansive and diverse settings. In this context, the potential of

AI emerges as a breakthrough technology for change and enhancement, with the possibility of revolutionizing various industries.

The challenge that is the subject of this paper lies in the limitations of traditional scaled agile development methods (SADMs) to solve the challenges associated with the management of large-scale projects effectively. Although agile methods have proven successful in small-team environments [1], they often run into issues when dealing with complex systems involving many stakeholders, large code bases, and distributed teams. As a result of this, organizations need innovative solutions to overcome these obstacles and optimize their development processes. We will try to address this problem with the help of AI, which has tremendous potential to address the challenges that SADMs (implementations) are facing. By using AI techniques, organizations can improve their efficiency, decision-making processes, and overall productivity, as well as automate repetitive tasks, optimize resource allocation, improve communication and collaboration, and gain insights from large amounts of data. In addition, machine learning (ML) algorithms can enable predictive analytics, allowing teams to anticipate and mitigate potential risks.

During the course of our research, we formulated the following research questions which will serve as our guidelines:

RQ1: How can AI-driven assistants be used effectively to address the most common challenges faced by SADMs in managing large-scale projects?
RQ2: What are the potential benefits and challenges of incorporating AI into SADMs?
RQ3: Which aspects of SADMs do AI-driven assistants improve?

The rest of the paper is structured as follows. We begin with an introduction to SAFe, which serves as our chosen representative of SADMs, upon which we focus exclusively. In Section 2, we provide an overview of related works in the field of large-scale agile development. In Section 3, we present an extensive overview of our research area, within which we conducted a systematic literature review (SLR) to narrow down the literature relevant to our specific area of interest. In Section 4, we present the findings of our SLR, and in Sections 4.1 and 4.2, we explore the challenges of SADMs as well as advantages and disadvantages of implementing AI within them. Section 5 is dedicated to classifying the challenges encountered during our research for the different configurations of SAFe, providing comprehensive descriptions of their scope, and shedding light on the different AI-driven assistants which can, to some extent, address their resolution. In Section 6, we try to respond to the research questions, and, finally, in Section 8, we conclude this paper, summarizing the key findings derived from our research.

Introduction to SAFe

Scaled agile development methods, such as SAFe [2] and LeSS [3], entail the application of agile principles on a larger scale, extending beyond small, co-located teams to multi-team efforts that involve numerous actors and interfaces with existing systems. These methods promote collaboration, change tolerance, adaptive software development, and active customer involvement while emphasizing rapid iterations and frequent feedback loops [4].

The most recent configuration of the SAFe framework, version 6.0 [2], is the foundation of knowledge between lean, agile, and DevOps for achieving business agility. It focuses on seven core business agility competencies that are essential to gaining and maintaining a competitive advantage in an increasingly digital world: lean–agile leadership, team and technical agility, agile product delivery, delivery of enterprise solutions, lean portfolio management, organizational agility, and continuous learning culture [2,5].

SAFe accommodates a wide range of development environments with four different configurations/levels, making it a versatile and adaptable approach for organizations of varying sizes and industries:

- **Essential SAFe—**The basic building block of other SAFe configurations. This level includes all the essential elements needed to benefit from SAFe and the starting point for any implementation. It provides complex solutions for large teams and it consists of a program and team level [2].

- **Large-Scale SAFe**—It provides complex solutions that do not require portfolio management assistance. It is used mainly by organizations with multiple Agile Release Trains (ARTs) working together [2].
- **Portfolio SAFe**—Intended primarily for companies to align agile development with value streams and ARTs. The concept of lean–agile budgeting empowers decision-makers and includes visibility of portfolio and Work In Progress (WIP) limits through the Kanban system, as well as objective metrics that support management and improvement through Bucket-size planning. This level of configuration consists of a portfolio, program, and team level [2].
- **Full SAFe**—A complete SAFe framework that includes all levels of SAFe—team, program, large-solution, and portfolio level. It includes a complete set of roles, events, and artifacts, and is suitable for complex organizations that need to synchronize ARTs across the entire enterprise [2].

One of the reasons SAFe is a preferred choice is its popularity [1] within the agile development landscape, evident from the amount of literature and resources dedicated and connected to it. This extensive knowledge base makes it easier for organizations to access information and support, as well as educate themselves on the experiences, success, and failure factors in other real-life applications. In the upcoming discussion, we will narrow our focus to SAFe as a representative example of SADMs, to explore its principles and benefits in connection to AI in greater depth.

2. Related Work

Throughout our investigation of challenges within SADMs, we encountered diverse publications that addressed the challenges we have consolidated in various contexts (see Section 4.1). Certain publications [6–8] conducted SLRs to identify challenges in SADMs systematically, while others [4,8–12] gleaned insights from practical experiences by conducting structured interviews and reporting their empirical findings. Additionally, specific studies focused on categorizing these challenges into distinct facets, including aspects such as stakeholder management and reoccurring concerns and patterns [6,11].

In our exploration of the advantages of implementing AI in SADMs, we encountered a variety of publications. Notably, we observed some publications [13,14], that either introduced or documented experiences related to pair-programming. Furthermore, we came across a paper that conducted an SLR [15], to provide an overarching view of the research domain, and to identify potential areas where AI-driven assistants could be deployed to offer assistance.

Uludag et al. [6] comprehensively identified 79 challenges arising from large-scale agile development, with 41 new challenges emerging and 38 existing ones being strengthened by large-scale agile development. Their proposed future work [4] involves leveraging their large-scale agile pattern language to address recurrent challenges, without specifying particular tools. Sinha et al. [7] identified 11 challenges categorized as internal and external factors. Their research, as stated in their paper as a point of future work, was used as a basis to investigate the practices to overcome the identified barriers and utilize the success factors identified. Ciancarini et al. [8] conducted an in-depth multi-vocal literature review supplemented by empirical investigation, cautioning about the limited generalizability of their findings due to a small sample size and observational nature. Despite not suggesting specific solutions, they recommended suggestions for further investigation. Kieran et al. [9] outlined nine challenges across 13 cases and offered recommendations for resolution, acknowledging the complexity of complete elimination due to multifaceted factors, without mentioning AI-driven assistant solutions. Kasauli et al. [10] derived their findings from qualitative interviews, presenting 24 challenges, and proposed solutions drawing from SAFe, LeSS, and their case companies, highlighting literature gaps and suggesting relevant solution candidates. Putta et al. [11] surveyed 204 practitioners to identify reasons, benefits, and challenges in adopting SADMs, refraining from offering specific solutions while emphasizing the need for future focus on quality assessment.

These sources significantly influenced our research, culminating in the synthesis of knowledge, particularly pertaining to challenges in large-scale development. Our contribution manifests in the proposition of a set of AI-driven assistants aimed at tackling these challenges to a certain extent. Interestingly, within the literature defined during the SLR process, no papers were encountered that **specifically** tackle challenges in large-scale development utilizing AI-driven assistants. This absence might be attributed, in part, to the limitations outlined in our research, as detailed in Section 7.

3. Research Method

In the context of SAFe, this paper aims to investigate and evaluate the current state of AI-driven assistants. Through a thorough assessment of related literature, we aim to highlight current research, improvements, and applications of AI in supporting agile techniques in large-scale software development environments. By analyzing the strengths and weaknesses of these AI-driven assistants critically, we aim to discover all the possibilities for development and create new opportunities for further studies.

The research method used was a systematic literature review (SLR), which will provide a summary of relevant material accessible from the digital libraries listed in Table 1. Employing the systematic review methodology outlined by Kitchenham and Charters [16], we ensured a meticulous and exhaustive approach throughout our research process. Based on the results obtained using this method, we will answer the research questions presented in Section 1.

Table 1. Chosen digital libraries.

Digital Library	Web Address
IEEE Xplore	http://ieeexplore.ieee.org/
ScienceDirect	https://www.sciencedirect.com/
SpringerLink	https://link.springer.com/
ACM Digital Library	https://dl.acm.org/

After forming the selection of digital libraries, we defined a set of search strings for the purpose of addressing the defined research questions. The selected search strings are shown in Table 2.

Table 2. Search strings.

Search String	Purpose
»scal*« AND »agile« AND (»SAFe« OR »Scaled Agile Framework«) AND (»challenge*« OR »difficult*« OR »problem*«)	Explore the challenges of the selected scaled agile development method. We address **RQ1** and **RQ3** partially.
»scal*« AND »agile« AND (»AI« OR »artificial intelligence« OR »ML« OR »machine learning«) AND (»advantage*« OR »improv*« OR »benefi*« OR »disadvantage*« OR »challeng*« OR »difficut*«)	Research ways of using AI in the field of SADM with a specific focus on the potential benefits and drawbacks. We address **RQ2**.
»scal*« AND »agile« AND (»AI« OR »artificial intelligence« OR »ML« OR »machine learning«) AND (»tool*« OR »techniq*« OR »agent*« OR »assist*«)	Explore specific assistants, techniques, or tools based on AI already implemented into SADM. We address **RQ3**.

We also defined inclusion and exclusion criteria, which were crucial for achieving a thorough assessment in this paper; they served as key filters that guided the selection of relevant research that matches our field of interest from the wide range of published literature. The set inclusion and exclusion criteria are shown in Table 3.

Table 3. Inclusion and exclusion criteria.

Inclusion Criteria	Exclusion Criteria
The full text is available in the selected digital libraries	The paper is not written in English
The paper was published in the last 5 years	Not relevant to the set research questions
Is related to AI assistants that are based on the latest technological advances	Tutorials, slides, presentations, and summaries.
Peer-reviewed literature	

With the defined criteria, we began the SLR process. The literature review was conducted in mid-2023 and consisted of six stages, which are presented in Figure 1.

1. **Initial search.**
   In the first phase of the SLR process, we obtained **2159** items. In the digital libraries ScienceDirect, SpringerLink, and ACM Digital Library, we selected papers in the fields of Computer Science, Informatics, and Software Engineering, and we filtered the search strings according to the set inclusion criteria.
2. **Duplicate removal.**
   The process of duplicate removal followed, where all results were normalized in a single literature list and duplicates were eliminated. We identified 313 instances where papers showed up multiple times, and, after their removal, the literature list was reduced to **1846** items.
3. **Title- and keyword-based screening.**
   In this stage, we eliminated the largest amount of papers; out of 1846 items, we reduced the literature list to **112** items. This significant change is attributed to the ever-growing interest in AI as a compelling research domain, leading to an influx of papers and case studies across various sectors like medicine and construction. In our literature selection, we deliberately limited our focus to the field of SADMs or the domains of Computer Science and Information Systems that implement them.
4. **Abstract-based screening.**
   The wider range of papers was additionally analyzed according to the abstract of the paper. Our literature list was reduced from 112 to **35** items.
5. **Content-based screening.**
   In this stage, we carried out a thorough review of the selected papers based on their content. We analyzed 35 papers and eliminated those that did not address our set field of research. At this stage, **24** items remained in the literature list.
6. **Snowballing.**
   In addition, we reviewed references of the selected papers to identify additional sources. This way, we were certain that we also included sources that were not found in the selected databases, but met the set criteria and addressed our research area. At this stage, we added six items, and therefore, this is how we ended up with **thirty** primary sources (see Section 4).

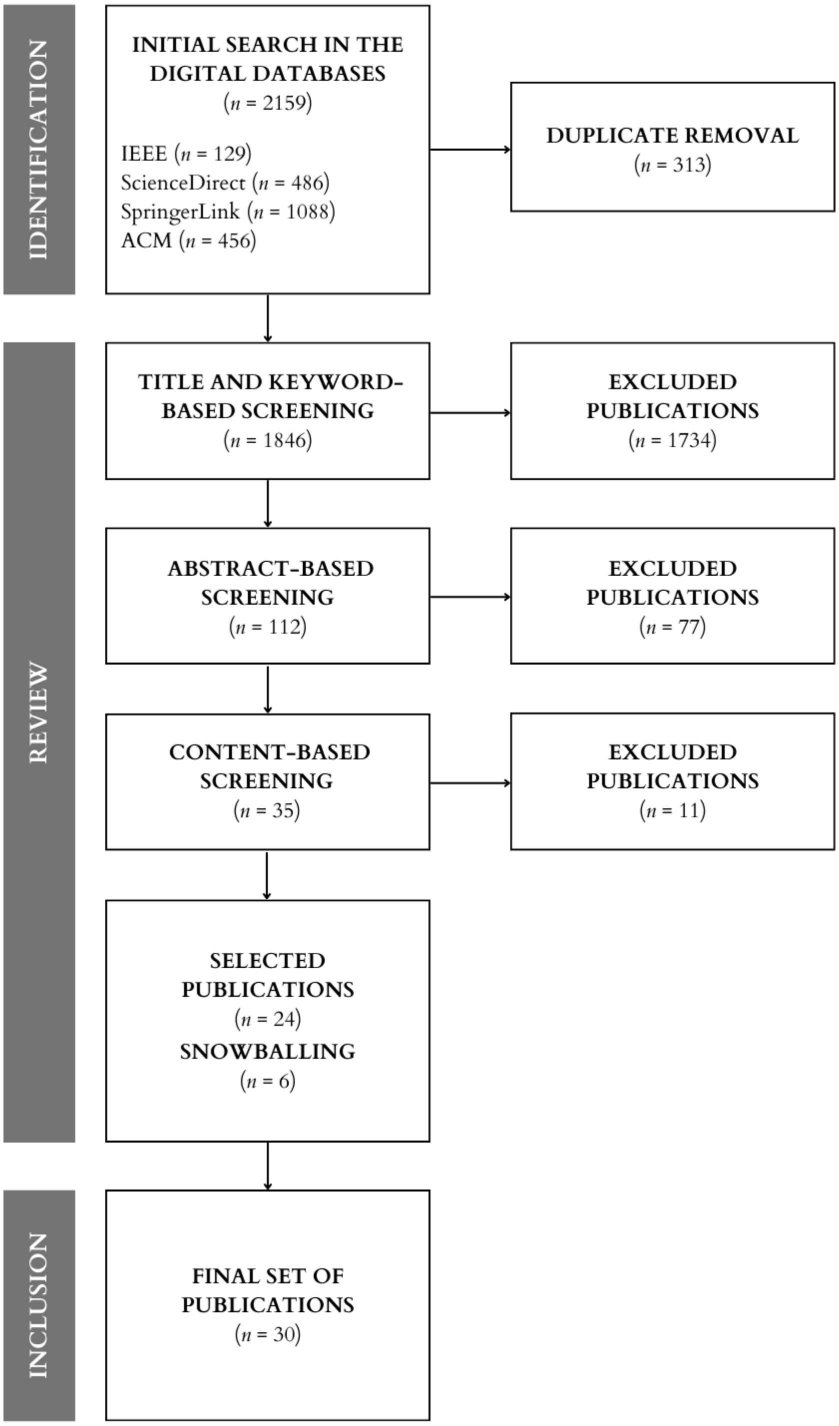

Figure 1. Systematic literature review process.

4. Review Results

From the primary literature list of 30 papers, we identified **15** papers that addressed the use of AI-driven assistants or were use cases on their own. Within this selection, we were interested in the specific usage domains of AI assistants in SADMs (see Table 4). The largest proportion of the reviewed literature presented in Table 5 addressed human assistance. Others included risk prediction, problem solving, and organizational success and efficiency.

Table 4. Usage domains of AI assistants.

Usage Domain	Sources
Human assistance	[9,11–14,17–26]
Risk prediction	[7,11,12,19,23,27–29]
Problem solving	[7,9,14,21,23,24,26]
Organizational success and efficiency	[10,14,15,27,30]
Organizational creativity	[7,10,22,23,30]
Development	[23,25,27,29,31]
Planning	[17,23,24,27]

Table 5. Primary literature list.

ID	Title	Authors	Source	Identified Assistant
1	A cautionary tale about the impact of AI on human design teams	Zhang et al.	[24]	✓
2	A classification and review of tools for developing and interacting with machine learning systems.	Mosqueira-Rey et al.	[29]	✓
3	A taxonomy of scaling agility	Limaj et al.	[32]	
4	AI Information Architecture	Hechler et al.	[33]	✓
5	AI-boosted software automation: learning from human pair programmers	Peng et al.	[13]	✓
6	Applications of ML/AI for Decision-Intensive Tasks in Production Planning and Control	Elbasheer et al.	[15]	✓
7	Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making	Jarrahi	[14]	✓
8	Artificial intelligence capability: Conceptualization, measurement calibration, and empirical study on its impact on organizational creativity and firm performance	Mikalef et al.	[30]	✓
9	Artificial Intelligence Technology	Huawei Technologies Co.	[28]	
10	Building the AI-Powered Organization	Fountaine et al.	[34]	✓
11	Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review	Edison et al.	[18]	
12	Computer-aided mind map generation via crowdsourcing and machine learning	Camburn et al.	[22]	✓
13	Decoding the agility of artificial intelligence assisted human design teams	Song et al.	[26]	✓
14	Documenting recurring concerns and patterns in large-scale agile development.	Uludağ et al.	[4]	
15	Empowering Software Engineering with Artificial Intelligence	Dam.	[17]	✓
16	Hybrid Intelligence	Dellermann et al.	[20]	✓
17	Identifying and Structuring Challenges in Large Scale Agile Development based on a Structured Literature Review	Uludag et al.	[6]	
18	Implementing Large-Scale Agile Frameworks: Challenges and Recommendations	Kieran et al.	[9]	
19	Issues in the adoption of the scaled agile framework.	Ciancarini et al.	[8]	
20	Needs and challenges for a platform to support large-scale requirements engineering: a multiple case study.	Fucci et al.	[12]	
21	Predicting failures in agile software development through data analytics	Batarseh et al.	[19]	✓
22	Requirements engineering challenges and practices in large-scale agile system development	Kasauli et al.	[10]	
23	Scaled Agile Framework Implementation in Organizations’, its Shortcomings and an AI Based Solution to Track Team’s Performance	Upasana et al.	[27]	✓
24	Strategic Challenges for Platform-based Intelligent Assistants	Zimmermann et al.	[25]	
25	SWOT: Strength, Weaknesses, Opportunities, and Threats for Scaling Agile Methods in Global Software Development	Sinha et al.	[7]	

Table 5. *Cont.*

ID	Title	Authors	Source	Identified Assistant
26	The AI Effect: Working at the Intersection of AI and SE	Carleton et al.	[23]	✓
27	The Future of Software Engineering: Where Will Machine Learning, Agile, and Virtualization Take Us Next?	Mancl et al.	[31]	
28	Tools and Algorithms for the Construction and Analysis of Systems	Fisman et al.	[35]	
29	Toward Hybrid teams: a platform to understand human-computer collaboration during the design of complex engineered systems	Song et al.	[21]	
30	Why Do Organizations Adopt Agile Scaling Frameworks? A Survey of Practitioners	Putta et al.	[11]	

The categorization of the AI assistants mentioned in Table 4 underscores their versatility, indicating that they are not limited to a single specific domain of application, but are designed to serve a broad range of purposes. As a result, organizations looking to integrate AI-driven assistants can frequently derive benefits across multiple dimensions rather than solely within their primary area of focus. In Table 6, we have summarized the selected AI-driven assistants, which are intended to assist us in matching potential solutions to the challenges within the context of SADMs that we have identified.

In the course of our research, we encountered papers [7,17,20,23,25,29–31,34,35] in which authors offered valuable recommendations and suggestions for the implementation of AI-driven assistants. Certain papers created new assistants tailored for specific domains, while others drew conclusions from empirical research, and arranged them in a more structured fashion. All of this was performed with the ultimate goal of making the knowledge obtained from these sources accessible to organizations and businesses seeking to enhance or adapt their operational efficiency.

Table 6. Identified AI-driven assistants.

ID	Source	Description
1	[24]	Investigation of the influence of deep learning AI on distributed human design teams in the field of engineering design, complemented with a human subject study that includes an abrupt problem change.
2	[29]	Categorization of AI-driven assistants, with a particular emphasis on the realm of ML. The categorization is structured in alignment with the developmental stages of ML systems, and it includes instances of AI-driven assistants within each distinct category.
4	[33]	Exploring use cases showcasing AI's application across various aspects of an information system and their relevance in enterprise contexts.
5	[13]	Exploring the challenges prevalent in software development settings and suggesting a cooperative approach involving both humans and AI (pair-programmers) to efficiently address them.
6	[15]	Uncovering common themes and trends in AI/ML-enabled manufacturing systems, and mapping the diverse application areas of AI/ML in Production Planning and Control (PPC) while also investigating its role in improving decision making.
7	[14]	Exploring how humans and AI can synergistically contribute their unique strengths in decision-making processes within organizations marked by uncertainty and complexity.
8	[30]	Identification of AI-specific resources, creating an assessment tool for measuring AI capabilities, and investigating the link between AI capability and organizational creativity and performance.
10	[34]	Organizing AI for scale, providing insights into common obstacles and how to communicate the shift to AI to everyone involved.
12	[22]	Introducing a method for crowd-sourcing design concepts and their hierarchical organization, by combining human evaluators and ML to streamline the creation of mind maps.
13	[26]	Investigating the impact of integrating AI-driven assistants into human teams for enhancing team agility, proving improved coordination, communication, and overall performance.
15	[17]	Harnessing the application of advanced AI machine learning techniques to create data-driven, automated approaches for software effort estimation, code patch development, and risk prediction within contemporary software development environments.

Table 6. *Cont.*

ID	Source	Description
16	[20]	Examining how the integration of human and AI collaboration, known as hybrid intelligence, enhances real-world business applications.
21	[19]	Introducing "analytics-driven testing (ADT)" that predicts software failures in agile sprints by applying analytical and statistical techniques, while also estimating error locations with a specified statistical confidence level.
23	[27]	Addressing challenges within SAFe and proposing a machine learning model as a potential solution to track team performance.
26	[23]	Delving into the synergy between AI and SE, investigating AI's contributions to SE and strategies for enhancing the development of AI systems by software engineers.

The classification of AI-driven assistants within the realm of scaled agile environments, presented in Table 4, serves a dual purpose. Firstly, it assists us in organizing their application across areas where companies and organizations often encounter challenges. Secondly, in the upcoming section, we will delve deeper into the details of these challenges, pinpointing their connection to distinct levels within SAFe, and try to bridge the gap between the most common challenges and their potential solutions. This approach aims to offer a potential strategy to address the unique challenges faced at each SAFe level through the use of AI-driven assistants.

4.1. Challenges in Scaled Agile Development

The use of SADMs is becoming increasingly common in modern software development organizations [1]. The transition to scaled agile development has many potential benefits, but there are also difficulties due to coordination challenges, communication difficulties, and lack of flexibility [9], among many other things. Many turn to the use of SADMs such as SAFe to overcome these problems, since they have established workflow patterns and processes, and are supported widely by extensible tools. However, empirical research on the adoption of these methods, their use, success, and problems, is still quite immature [4,6,7,9,11,36]. Kieran et al. [9] conducted a study analyzing 13 large projects of different global companies over a period of 15 years. Their results show that the success of implementing SADMs depends on many different factors, not only following the framework's regulations "by-the-book". Some businesses and organizations continued to use the selected method and, over time, observed bigger success; others switched to a different type of method, and consequently enhanced their performance and efficiency; and, finally, some totally abandoned the use of the SADMs for several reasons. Figure 2 displays a graphic representation of SADMs' implementation over time.

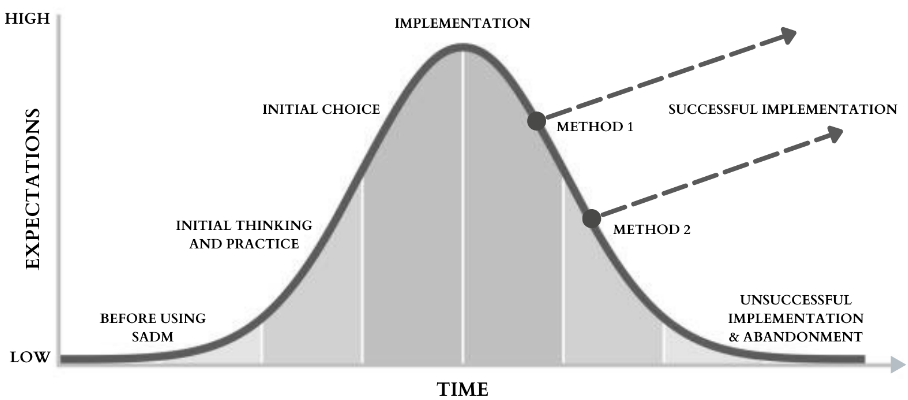

Figure 2. The adoption of large-scale agile frameworks (based on [9]).

A 15-year study [9], a 2021 survey [10], and a proposition for the implementation of an AI-driven solution [27] highlight common challenges in organizations adopting SADMs.

Uludag et al. [6] conducted an SLR that identified stakeholders in SADMs. The review pinpointed 79 challenges, categorized into 11 groups.

Putta et al. [11] explored the reasons for adopting SADMs, the potential benefits, as well as the satisfaction their implementation can cause in scaled environments, through a survey of software practitioners. The gap of documenting reoccurring concerns and patterns in large-scale development is one of the key points that Uludag et al. [4] were trying to address in their paper. They introduced their own pattern language, which, equipped with structured interviews with 14 large-scale agile development experts from 10 organizations, gave us additional insights as to what challenges and concerns real-life organizations face.

Sinha et al. [7] conducted an SLR and a SWOT analysis exploring the effects of the adoption of SADMs. In addition to this, Ciancarini at al. [8] undertook an SLR complemented by empirical research to gain a deeper understanding of adopting SADMs, with a focus on SAFe. The input data were gathered from 25 respondents across 17 companies in eight countries, highlighting challenges related to decision making, organizational structures, and technical and managerial competencies. Separate case studies with three different companies from different industries were the topic of Fucci et al.'s [12] research paper. These studies centered on identifying current issues, needs, and implementation challenges through structured interviews.

In Figure 3, we summarize the consolidated list of the identified challenges in large-scale development environments based on the SLR.

When discussing the array of challenges (see Figure 3), it is essential to note that we are not addressing any significant ones explicitly. It is also important to highlight that these challenges vary in terms of their characteristics, including their nature and category, as well as their relevance to specific domains. While our list may not cover all challenges encountered by organizations implementing SADMs, we aim to provide the most accurate representation of the real-world circumstances, recognizing the limitations outlined in Section 3. We proceed to explore the set of identified challenges in Section 5, during which we leverage the capabilities of AI to provide recommendations for their resolution.

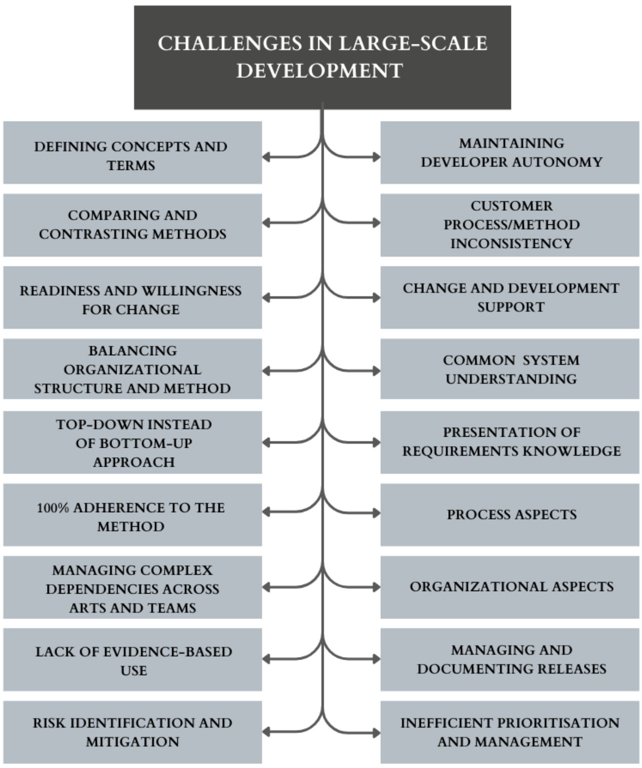

Figure 3. Identified challenges during SADM implementation.

4.2. Advantages and Disadvantages of Implementing AI in SADMs

Incorporating AI into software development processes yields numerous benefits, including enhanced productivity, quality, and creativity across multiple stages of the developmental process [13–15,30]. As stated in [36], shorter product life cycles, fragmented value chains, and novel organizational structures like value creation networks, platform solutions, and cluster organizations have surfaced, frequently associated with innovative technologies such as AI, ML, and Blockchain. The automation of repetitive tasks using AI-driven assistants, like code reviewing, testing, and error detection, offers significant savings in manual labor and reduces development cycles [13]. AI has the capacity to identify patterns and trends via advanced data analysis, empowering developers to make informed decisions and anticipate potential issues. Moreover, the integration of AI-driven algorithms presents opportunities to enhance code performance, suggest upgrades, tackle documentation generation, streamline communication, and promote knowledge exchange within development teams.

Peng et al. [13] examined how AI can be applied to automate software development by using pair programmers, which consist of a developer and an AI-driven assistant. They found that, while there are potential advantages to incorporating AI in the development process, there are also various challenges. To address this, they suggest automating collaboration between humans and AI; developers would continue to follow established development methods, with the AI-driven assistant working in the background to provide support when developers encounter issues. This combination is expected to improve efficiency and quality, by reducing repetitive tasks and assisting new developers in thinking and working like experts. Elbasheer et al. [15] explored the domains of Planning and Control and the impact of AI assistants in this context. Through an SLR, they identified three key areas that benefit from the use of AI assistants.

The author of [14] conducted research that explores the distinctions between human thought processes and AI assistant procedures, where the author ultimately concluded that introducing AI assistants to help humans will lead to significant business success. Mikalef et al. [30] investigated the capabilities that arise when deploying AI assistants in large-scale settings. Their findings categorized these capabilities into three groups: tangible (e.g., data and technologies), human (e.g., encompassing technical and business skills), and intangible (e.g., the ability to coordinate and adapt), offering insights into the broader implications of implementing such technologies. Similar research was conducted by Fountaine et al. [34], whose findings highlighted common challenges associated with implementing AI, provided guidance on transitioning to its use, and pointed out the potential benefits that AI may have in the organizational setting it is implemented in. In Figures 4 and 5, we summarize the consolidated list of the identified advantages and challenges of implementing AI.

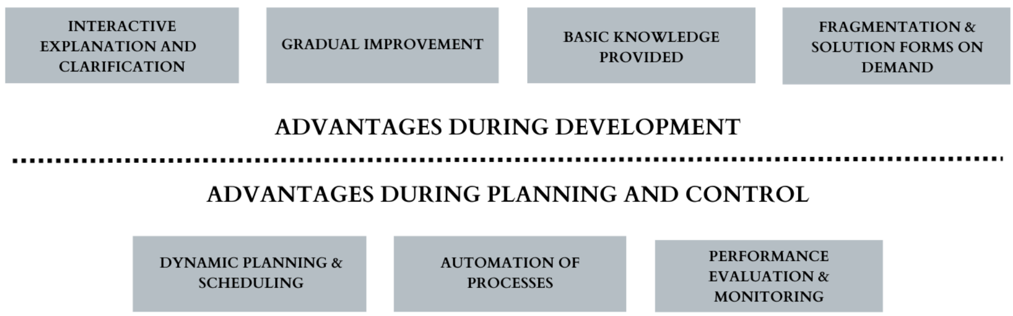

Figure 4. Identified advantages of implementing AI during SADM implementation.

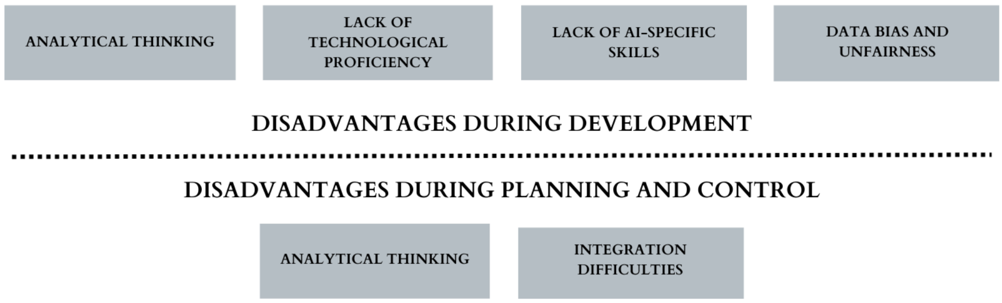

Figure 5. Identified disadvantages of implementing AI during SADM implementation.

5. Applying Review Results to SAFe Environments

As stated in the introduction of this paper, our goal is to demonstrate a connection between the challenges we have identified and the structural organization (different levels) of SAFe (see Table 7). Our aim is to pinpoint the precise locations within SAFe where these obstacles are most likely to occur. The compilation of all recognized challenges within every level of SAFe incorporates a thorough exploration of the multitude of (possible) challenges existing at the full level. This is warranted, given that the full level, as the highest level of the SAFe hierarchy, encompasses all its subordinate parts (essential, large-solution, and portfolio).

A comprehensive description of the scope and nature of these challenges will be presented in the following sections. Furthermore, the research will offer insights into suggestions for AI-driven assistants potentially proficient in addressing these challenges, along with suggestions and recommendations for their implementation.

Table 7. Identified challenges in SAFe and the level in which they occur.

Level	Identified Challenge
Essential (E)	E1: Defining concepts and terms E2: Readiness and willingness for change E3: Maintaining developer autonomy E4: Customer process/method inconsistency E5: Presentation of requirements' knowledge E6: Process aspects E7: Risk identification and mitigation
Large Solution (LS)	*E1–E7 +* LS1: Change and development support LS2: Common system understanding LS3: Managing and documenting releases LS4: Managing complex dependencies across ARTs and teams
Portfolio (P)	*E1–E7 + LS1–LS4 +* P1: Comparing and contrasting methods P2: Balancing organizational structure and method P3: Top-down instead of bottom-up approach P4: 100% adherence to the method P5: Lack of evidence-based use P6: Organizational aspects P7: Inefficient prioritization and management

5.1. Challenges at the Essential Level

E1: Defining concepts and terms—The fundamentals are well-defined in the papers that introduced methods like SAFe and LeSS [3]. However, when these methods are applied in different contexts, the guidelines for their implementation can become less clear [9]. Many companies cite misconceptions about the concepts and procedures within these methods, along with substantial variations in their interpretation and application across different areas. A recurring issue that we have encountered during our research is the presence of abstract terminology. The lack of detailed explanations can pose a significant challenge for organizations attempting to transition to SADMs.

E2: Readiness and willingness for change—To transition to agile methods successfully, organizations and employees must be open to change. While employees may embrace changing software processes, they might not be ready to adopt specific methods. Frameworks like SAFe and LeSS [3] offer structures and processes, but often lack guidance on assessing overall readiness for agile transformation at scale [9]. In their research, the authors of [7] found considerable skepticism towards adopting agile development. Many developers have long been accustomed to the traditional sequential (waterfall) approach, making the shift to a completely new way of working uncomfortable and prone to organizational challenges. Approximately 20% of papers in [7] highlighted the absence of an agile process evaluation mechanism as a significant concern. In [14,34], we come across the appearance of employees' resistance, since there is an ever-growing concern that AI-assistants will replace them fully (instead of adapting to implementing them in their workflows).

E3: Maintaining developer autonomy—Autonomy in scaled environments is being recognized as an increasingly significant challenge. However, SADMs furthermore deteriorate this issue, introducing additional constraints and inflexibilities. According to the authors of [9], there have been instances where developers' proposals for tool and process improvements were rejected due to perceived incompatibility with the new method's implementation. Furthermore, cultural and linguistic difficulties were noted in [7], particularly when development teams are spread across different countries.

E4: Customer process/method inconsistency—Implementing a (new) SADM presents a significant challenge, since it is more difficult to transform the organization's predetermined processes and structures. In certain cases, as noted in [9], companies have had to establish collaborative agreements with clients, outlining how they will work together to produce software within a specific development framework. The authors of [10] note that, despite close customer relationships, there can be a considerable gap between customers and developers. This disconnect may arise because development teams struggle to empathize with the customer's perspective and articulate how their work benefits them directly. Additionally, as per [10], crafting user stories that provide value to the customer directly can be particularly challenging. Such stories may be too intricate to complete and demonstrate within a single sprint or iteration. Maintaining reusable customer insights within intricate product families is equally problematic. Consequently, any changes may necessitate work repetition to obtain similar customer-related information [10].

E5: Presentation of requirements' knowledge—In this context, challenges, as outlined in [10], include issues such as managing people in hierarchical layers instead of an organizational decomposition, and setting appropriate thresholds for requirements. In addition to this, we drew insights from [12], where the authors exposed challenges such as coping with information overload (dealing with the collection, search, and analysis of vast amounts of information), coping with limitations of the chosen development assistants, and managing dependencies between requirements.

E6: Process aspects—This challenge encompasses various aspects that identify common issues within the development processes of SADMs directly. These challenges include:

– prioritizing the highest-priority tasks [10];
– determining requirements' completeness [10];
– managing requirements using different processes, tools, and levels of detail [10];
– determining a clear quality threshold (for requirements, products and deliverables) that signifies readiness for release [10].

In addition to this, Upasana et al. [27] mentioned the lack of teamwork, excessive commitment, project abandonment, and decreased motivation among development teams, while Sinha et al. [7] observed a visible lack of interpersonal communication, agile coaching, and linguistic differences.

E7: Risk identification and mitigation—The challenges that belong in this category mostly address the fact that it is difficult to identify and assess risks, as well as recognize the potential threats they can have on a project's timeline, overall quality, and finances [6]. Due to the inherent unpredictability, temporal relationships, and particularly the dynamic nature of software, predicting future risks is very difficult [17].

Potential solutions and suggestions for implementing AI for the challenges identified at the essential level are presented in Table 8.

Table 8. Suggestions for the resolution of challenges at the essential level.

Challenge	AI-Driven Solution	Suggestions and Recommendations for Implementing AI
E1: Defining concepts and terms	[20]	[7,17,23,34,35]
E2: Readiness and willingness for change	/	[14,20,33,34]
E3: Maintaining developer autonomy	[20]	[17,23,30,34]
E4: Customer process/method inconsistency	[27]	[14,23,31,34]
E5: Presentation of requirements' knowledge	[19]	[14,20,30]
E6: Process aspects	[21]	[14,17,23]
E7: Risk identification and mitigation	[15,19,29]	[14,17]

5.2. Challenges at the Large-Solution Level

LS1: Change and development support—In this context, the authors of [10] have identified three specific challenges:

– Management of experimental or poorly defined requirements.
– Synchronization of development—Due to the nature of large-scale development environments, synchronizing development activities among teams becomes complex, which limits agility and speed.
– Update of Requirements—Requirements defined at the beginning of a sprint often become outdated and no longer align with the solution.

Upasana et al. [27] highlight the management of unresolved dependencies. With many cross-functional teams relying on data and information, unresolved dependencies between data points can jeopardize progress. Fountaine et al. [34] recognized a number of issues when an organization tries to adapt from rigid and risk-averse to agile, experimental, and adaptable. Dam [17] discussed how software engineering's primary focus continues to be on addressing problems or adding new functionalities, instead of adapting to the ever-growing change.

LS2: Common system understanding—Kasauli et al. [10] noted that, in large-scale agile development environments, there is a notable absence of a shared understanding of the system. Various challenges within this context include insufficient documentation to support testing and stories, confusion between thinking at the system and component levels, and inadequate tracking and its upkeep, just to name a few.

LS3: Managing and documenting releases—Besides the technical aspects, this challenge addresses the organizational ones; there is a visible challenge regarding collaboration and communication between ARTs, development teams, and stakeholders [6]. Additionally, as the product or solution develops, it becomes crucial to keep a complete record of updates. Failure to do so can cause release misalignments or delays. In [6,7], creating lightweight documentation that contains all the necessary requirements is listed as a reoccurring challenge.

LS4: Managing complex dependencies across ARTs and teams—As organizations scale their solutions and procedures, in most cases, so do the dependencies between ARTs and teams. This challenge is often represented in a chain of codependent tasks, where one team's success depends on another. They can cause delays, reduce productivity, and even compromise the success of the project if managed improperly [6].

Table 9 provides a presentation of potential solutions and recommendations for the implementation of AI in addressing the challenges identified at the large-solution level.

Table 9. Suggestions for the resolution of challenges at the large-solution level.

Challenge	AI-Driven Solution	Suggestions and Recommendations for Implementing AI
LS1: Change and development support	[27]	[14,17,23,31,34]
LS2: Common system understanding	[14,20,23]	[17,30]
LS3: Managing and documenting releases	/	[14,29,35]
LS4: Managing complex dependencies across ARTs and teams	/	[7,14,23,29]

5.3. Challenges at the Portfolio Level

P1: Comparing and contrasting methods—Selecting the right SADM can be challenging for many companies, due to the absence of a comparative evaluation model. Kieran et al. [9] noted that many employees think the choice of a development method often seems ad hoc, and sometimes it is unclear where the decision originates. Survey results discussed by Ciancarini et al. [8] indicated that a significant number of respondents found the actual implementation of SADMs to be overly complex and challenging to grasp.
P2: Balancing organizational structure and method—The challenge in implementing SADMs arises from their predefined organizational structures, processes, and proprietary tools. These structures are evolving constantly in response to external competition and regulatory demands, making a one-size-fits-all approach challenging. In their research, Sinha et al. [7] documented a lack of support and commitment from the upper management.
P3: Top-down instead of bottom-up approach—Many implementations have taken either a purely bottom-up or top-down approach, rather than a combination of the two. Research [9] indicated that top-down approaches have yielded mixed results. Ciancarini et al. [8] highlighted a set of challenges associated with the adoption of SADMs. One major challenge is that their implementation often leads to top-down organizational control, resulting in a structure resembling a waterfall approach that lacks true agility and flexibility.
P4: 100% adherence to the method—When a formal method like SAFe is used, there is a common tendency to evaluate the success of agile transformation by how closely the organization adheres to the method rather than the value it actually delivers. It is observed that SADMs encounter difficulties or issues frequently during the final 20–30% of development activities, and an overwhelmingly large portion of the effort and stress (roughly 80–90%) is expended on achieving the last 5% [9]. Moreover, a notable issue highlighted in 30% of the selected papers in [7] is the lack of adequate agile training for scaled development environments.
P5: Lack of evidence-based use—There are limited empirical case studies that examine the practical application of prescriptive principles in SADMs. According to organizations' reports [9], they encountered situations where employees faced significant challenges, and they struggled to find relevant information within the documentation of the method they were using in regard to their solution.
P6: Organizational aspects—In [10], years of research have uncovered three prevalent challenges within organized scaled environments:

- Disparity between plan-oriented, document-heavy approaches at the system level and the value-driven, agile approaches at the team level.
- Requirement-based validation and verification processes, which are inherently incomplete and incremental.
- Infrastructure and the critical aspect of timely upgrades.

Sinha et al. [7] noted that many scaled development environments also grappled with the issue of uneven task allocation. Since management assigns tasks to teams, there is the potential for tasks to be distributed unevenly among team members. In [23,34], we came across the long-abiding issue that is "quick wins". They address the fact that organizations need to focus on developing a portfolio of initiatives for a longer period of time, so that they can maximize their return on investment.

P7: Inefficient prioritization and management—This challenge combines issues like difficulties in defining clear and visible priorities that are in line with the organizations' strategic goals and dealing with loss of management control [6]. This may be due to the fact that there is a lack of automated help for effort estimation [17]. Failure to address these issues effectively can cause poor allocation of resources, which might lead to delays, misalignments, additional costs, and cancellation of the project(s) [12].

In Table 10, we have compiled a list of potential solutions and recommendations for the integration of AI in addressing the challenges identified at the portfolio level.

Table 10. Suggestions for the resolution of challenges at the portfolio level.

Challenge	AI-Driven Solution	Suggestions and Recommendations for Implementing AI
P1: Comparing and contrasting methods	/	[7,14]
P2: Balancing organizational structure and method	[13]	[14,20,23]
P3: Top-down instead of bottom-up approach	/	[22,23]
P4: 100% adherence to the method	[13]	[20,22]
P5: Lack of evidence-based use	/	[7,22,23]
P6: Organizational aspects	[13,27]	[20,22]
P7: Inefficient prioritization and management	[13,15]	[24,29,30]

6. Discussion

This paper presents a comprehensive examination of integrating AI-driven assistants within SADMs. We have identified and emphasized a range of significant findings and insights resulting from our chosen research method—SLR. Our research focused primarily on the incorporation of AI-driven assistants into a specific SADM, known as SAFe, with the aim of comprehending the potential advantages and implications of their utilization in software development.

Firstly, we investigated how AI-driven assistants can address the common challenges encountered by SADMs when managing large-scale projects effectively. Secondly, we delved into the potential benefits associated with the integration of AI in SADMs. Lastly, we explored how AI-driven assistants can enhance specific aspects of SADMs.

RQ1: ***How can AI-driven assistants be used effectively to address the most common challenges faced by SADMs in managing large-scale projects?***

In our research, we have found that AI-driven assistants hold significant promise in addressing the challenges encountered in managing large-scale projects within SADMs. As presented in Section 4.1, the consolidated list of challenges we identified in the literature scope is substantial. Examples of such challenges include balancing organizational structure and method, maintaining developer autonomy, common system understanding, etc. However, it is important to note the positive impact that the integration of AI-driven assistants can have in the solution of these challenges. Given the adaptability of SAFe to diverse organizational sizes, our initial aim was to categorize the identified challenges according to the SAFe levels (essential, large-solution, and portfolio) where they are most likely to occur.

Our SLR unveiled a multitude of papers (citation rates defined in Figure 6) that either tackled these challenges explicitly through the deployment of specialized AI-driven assistants, or provided overall guidance on how organizations could address these challenges effectively by integrating AI. In Section 5, we demonstrate a connection between the identified challenges, their potential solutions involving AI-driven assistants, and recommendations for the implementation of AI. This comprehensive analysis encompasses all identified challenges at every SAFe level, with the complete set of challenges spanning the entirety of the SAFe hierarchy, including the full level, which is the highest and most extensive one.

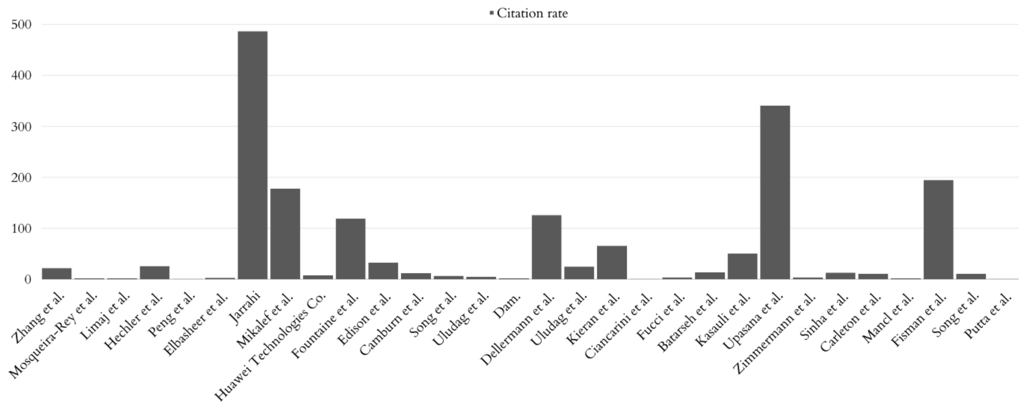

Figure 6. Citation rates of the acquired literature based on Web of Science as of December 2023. [4,6–15,17–25,27–35]

RQ2: ***What are the potential benefits and challenges of incorporating AI in SADMs?***

In the course of our research, we placed particular emphasis on identifying the advantages and challenges associated with the integration of AI within the context of SADMs. The results of our research, which relied on the limited literature gathered during the SLR process, are presented in Section 4.2. During our research, we identified seven distinct advantages, encompassing aspects such as process automation, dynamic planning and scheduling, as well as iterative enhancement, among others. In addition to this, we have delineated six challenges, spanning from critical thinking, biases and unfairness in data, and challenges in integration. Our analysis, supported by empirical studies, underscores that the integration of AI across a diverse spectrum of domains within SADMs has the potential to result in positive outcomes in terms of organizational performance and potential.

RQ3: ***Which aspects of SADMs do AI-driven assistants improve?***

Expanding upon the insights derived from the SLR findings, as presented in Section 4, our subsequent step involved categorizing the identified assistants, aimed at supporting organizations and companies in addressing specific challenges. In this subsection, we classified these assistants into the overarching category of *Usage Domains*, encompassing areas such as human assistance, risk prediction, problem solving, and various others.

Upon closer examination, we noticed an interesting finding; some assistants cross several sections within the category. This underscores the remarkable adaptability and applicability of AI-driven assistants, illustrating their capacity to address a wider range of challenges. Furthermore, we performed a categorization of the most common challenges based on different configurations of SAFe, which is presented in Section 5. This approach allows organizations to pinpoint challenges corresponding to the specific SAFe configuration they have implemented. Consequently, our categorization offers a different perspective, enabling organizations to deploy AI-driven solutions strategically and manage a wide range of challenges across their organizational structures effectively.

7. Limitations and Threats of Validity

In the process of the SLR, we deliberately shifted our focus to SAFe, as it stands out as the most widely adopted framework. The literature incorporated into our research drew mainly from practical experiences, success and failure stories within the context of SAFe, and the utilization of AI-driven assistants across various organizational domains. While the study extensively examines SAFe as the primary SADM, recognizing the unique characteristics of other agile frameworks could enhance the study's applicability and broaden its implications across different SADMs.

The AI-driven assistants and recommendations presented in Sections 5.1–5.3 are listed as potential solutions, yet they might not comprehensively address the breadth of challenges outlined. Given the expansive nature of these challenges, definitive assertions regarding their efficacy as absolute solutions remain ambiguous. Relying on literature and sourced claims for our insights, we assert a reasonably accurate estimation of their potential. Nevertheless, practical evaluation of their effectiveness or efficiency in authentic settings stands as an impractical undertaking at present. Our aim was to investigate how to (try to) solve the emerging challenges effectively with the help of the latest technological advancements. In regard to this, another limitation in our study is the absence of a structured classification or hierarchy of identified properties among AI-driven assistants. Our research primarily concentrated on the capacity of AI in addressing challenges within SADMs rather than extensively analyzing or categorizing these properties. Therefore, a comprehensive classification of the discovered properties was beyond the scope of our investigation.

It is important to note that our research and literature selection were conducted in mid-2023. Therefore, there is a possibility that recent innovations, unaccounted for in our research, may have an influence on our findings. Although we have spent considerable time and effort developing relevant search strings and conducting a structured database search, there is still some possibility that not all relevant papers have been identified. Additional literature was identified through a reverse search (Stage 6: Snowballing) of the analyzed papers in the literature search process. Despite our best efforts, there remains a possibility of overlooking a critical paper.

8. Conclusions and Future Work

This paper has delved into the realm of AI-driven assistants within the context of SADMs, especially SAFe, offering a comprehensive overview of their potential, categorization based on application domains, and the challenges they aim to address. Through the course of this paper, we have uncovered and provided detailed explanations of the challenges that are encountered commonly in SADMs, with a particular emphasis on SAFe. The challenges identified encompass issues related to coordination, resource utilization, and resistance to change, which are of considerable significance in large-scale software development.

Through the exploration of challenges articulated in the paper's introduction, we aim to propose AI-driven solutions that partially address these issues. To achieve this objective, an initial exploration involved assessing the potential advantages and disadvantages of integrating AI within SADMs. Our findings suggest a preponderance of advantages over disadvantages. Nonetheless, it is imperative for organizations and enterprises to exercise caution and informed consideration in the adoption of AI within their operational frameworks. We have also assembled a comprehensive selection of AI-driven assistants that are, to some degree, capable of solving the aforementioned challenges. By categorizing and evaluating the potential of these assistants, we are shedding light on their potential contributions to enhance different aspects of SAFe. Additionally, we have not only emphasized the technical potential of the set of AI-driven assistants, but we have also considered the organizational aspects that are essential to their successful implementation; we have curated a collection of recommendations and suggestions gleaned from extensive literature, emphasizing established best practices derived from real-world instances.

The findings derived from our research suggest several propositions for future work. We propose that forthcoming research should focus on identifying AI-driven assistants capable of enhancing all facets across diverse SAFe levels (e.g., within the portfolio level, to achieve Agile Product Delivery, which tools and assistants can improve Design Thinking and Lean UX). Conducting empirical studies within real-world SADM environments would offer invaluable insights into the practical effectiveness and challenges of implementing AI assistants. Furthermore, an exploration into the potential integration of emerging technologies like ML models or NLP to enhance the capabilities of these assistants represents a promising avenue for advancement.

In summary, the integration of AI-driven assistants carries the potential to reshape the way large-scale projects are designed and executed. Empirical data [13–15,30,36] during this research have shown that the adoption of such AI-driven solutions can result in increased levels of collaboration, enhanced decision-making processes, and optimal project outcomes, as businesses strive to remain innovative and competitive. However, according to the discoveries in this paper, achieving a lasting and effective integration of AI-driven assistants requires a well-rounded approach that considers the advantages as well as drawbacks of AI-integration.

Author Contributions: Conceptualization, V.S.; Methodology, V.S. and L.P.; Supervision, L.P.; Validation, V.S. and L.P.; Visualization, V.S.; Writing—original draft, V.S. and L.P.; Writing—review & editing, V.S. and L.P. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge financial support from the Slovenian Research Agency (Research Core Funding No. P2-0057).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. State of Agile Report. 2022. Available online: https://digital.ai/resource-center/analyst-reports/state-of-agile-report/ (accessed on 20 October 2023).
2. Leffingwell, D. Scaled Agile Framework. 2023. Available online: https://scaledagileframework.com/ (accessed on 18 July 2023).
3. Large Scale Scrum (LeSS). 2023. Available online: https://less.works/ (accessed on 18 July 2023).
4. Uludag, O.; Harders, N.M.; Matthes, F. Documenting recurring concerns and patterns in large-scale agile development. In Proceedings of the 24th European Conference on Pattern Languages of Programs Irsee, Germany, 4–8 July 2019. [CrossRef]
5. What Is SAFe Methodology? A Complete Guide to Scaled Agile Framework. 2023. Available online: https://www.invensislearning.com/blog/what-is-safe-methodology/ (accessed on 19 July 2023).
6. Uludag, O.; Kleehaus, M.; Caprano, C.; Matthes, F. Identifying and structuring challenges in large-scale agile development based on a structured literature review. In Proceedings of the 2018 IEEE 22nd International Enterprise Distributed Object Computing Conference, Stockholm, Sweden, 16–19 October, 2018.
7. Sinha, R.; Shameem, M.; Kumar, C. SWOT: Strength,weaknesses, opportunities, and threats for scaling agile methods in global software development. In Proceedings of the 13th Innovations in Software Engineering Conference, Jabalpur, India, 27 February 2020. [CrossRef]
8. Ciancarini, P.; Kruglov, A.; Pedrycz, W.; Salikhov, D.; Succi, G. Issues in the adoption of the scaled agile framework. In Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice, Pittsburgh, PA, USA, 21–29 May 2022; Volume 22, pp. 175–184. [CrossRef]
9. Kieran, C.; Noel, C. Implementing Large-Scale Agile Frameworks: Challenges and Recommendations. *IEEE Softw.* **2019**, *36*, 44–50.
10. Kasauli, R.; Knauss, E.; Horkoff, J.; Liebel, G.; de Oliveira Neto, F.G. Requirements engineering challenges and practices in large-scale agile system development. *J. Syst. Softw.* **2021**, *172*, 110851. [CrossRef]
11. Putta, A.; Uludag, O.; Hong, S.L.; Paasivaara, M.; Lassenius, C. 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), Bari, Italy, 11–15 October 2021. [CrossRef]
12. Fucci, D.; Palomares, C.; Franch, X.; Costal, D.; Raatikainen, M.; Stettinger, M.; Kurtanovic, Z.; Kojo, T.; Koenig, L.; Falkner, A.; et al. Needs and challenges for a platform to support large-scale requirements engineering: A multiple-case study. In Proceedings of the International Symposium on Empirical Software Engineering and Measurement, New York, NY, USA, 8–9 September, 2018. [CrossRef]
13. Peng, X.; Xing, Z.; Sun, J. AI-boosted software automation: Learning from human pair programmers. *Sci. China Inf. Sci.* **2019**, *62*, 1–3. [CrossRef]
14. Jarrahi, M.H. Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making. *Bus. Horiz.* **2018**, *61*, 577–586. [CrossRef]
15. Elbasheer, M.; Longo, F.; Nicoletti, L.; Padovano, A.; Solina, V.; Vetrano, M. Applications of ML/AI for Decision-Intensive Tasks in Production Planning and Control. *Procedia Comput. Sci.* **2022**, *200*, 1903–1912. [CrossRef]
16. Kitchenham, B.A. 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, EAST '12, New York, NY, USA, 22 September 2012. [CrossRef]
17. Dam, H.K. Empowering Software Engineering with Artificial Intelligence. *Lect. Notes Bus. Inf. Process.* **2019**, *367*, 22–32. [CrossRef]
18. Edison, H.; Wang, X.; Kieran, C. Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review. *IEEE Trans. Softw. Eng.* **2022**, *48*, 2709–2731. [CrossRef]
19. Batarseh, F.A.; Gonzalez, A.J. Predicting failures in agile software development through data analytics. *Softw. Qual. J.* **2018**, *26*, 49–66. [CrossRef]

20. Dellermann, D.; Ebel, P.; Söllner, M.; Leimeister, J.M. Hybrid Intelligence. *Bus. Inf. Syst. Eng.* **2019**, *61*, 637–643. [CrossRef]
21. Song, B.; Zurita, N.F.S.; Zhang, G.; Stump, G.; Balon, C.; Miller, S.W.; Yukish, M.; Cagan, J.; McComb, C. Toward hybrid teams: A platform to understand human-computer collaboration during the design of complex engineered systems. *Proc. Des. Soc. Des. Conf.* **2020**, *1*, 1551–1560. [CrossRef]
22. Camburn, B.; Arlitt, R.; Anderson, D.; Sanaei, R.; Raviselam, S.; Jensen, D.; Wood, K.L. Computer-aided mind map generation via crowdsourcing and machine learning. *Res. Eng. Des.* **2020**, *31*, 383–409. [CrossRef]
23. Carleton, A.D.; Harper, E.; Menzies, T.; Xie, T.; Eldh, S.; Lyu, M.R. The AI Effect: Working at the Intersection of AI and SE. *IEEE Softw.* **2020**, *37*, 26–35. [CrossRef]
24. Zhang, G.; Raina, A.; Cagan, J.; McComb, C. A cautionary tale about the impact of AI on human design teams. *Des. Stud.* **2021**, *72*, 100990. [CrossRef]
25. Zimmermann, A.; Schmidt, R.; Sandkuhl, K. Strategic Challenges for Platform-based Intelligent Assistants. *Procedia Comput. Sci.* **2020**, *176*, 966–975. [CrossRef]
26. Song, B.; Zurita, N.F.S.; Gyory, J.T.; Zhang, G.; McComb, C.; Cagan, J.; Stump, G.; Martin, J.; Miller, S.; Balon, C.; et al. Decoding the agility of artificial intelligence-assisted human design teams. *Des. Stud.* **2022**, *79*, 101094. [CrossRef]
27. Ameta, U.; Patel, M.; Sharma, A.K. Scaled agile framework implementation in organizations', its shortcomings and an AI based solution to track team's performance. In Proceedings of the 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 7–9 October 2022.
28. H.T.C. Ltd. *Artificial Intelligence Technology*; Huawei Technologies Co., Ltd.: Shenzhen, China, 2023. [CrossRef]
29. Mosqueira-Rey, E.; Pereira, E.H.; Alonso-Ríos, D.; Bobes-Bascarán, J. A classification and review of tools for developing and interacting with machine learning systems. *Proc. ACM Symp. Appl. Comput.* **2022**, *4*, 1092–1101. [CrossRef]
30. Mikalef, P.; Gupta, M. Artificial intelligence capability: Conceptualization, measurement calibration, and empirical study on its impact on organizational creativity and firm performance. *Inf. Manag.* **2021**, *58*, 103434. [CrossRef]
31. Mancl, D.; Fraser, S.D. The Future of Software Engineering: Where Will Machine Learning, Agile, and Virtualization Take Us Next? *Lect. Notes Bus. Inf. Process.* **2021**, *426*, 222–230. [CrossRef]
32. Limaj, E.; Bernroider, E.W. A taxonomy of scaling agility. *J. Strateg. Inf. Syst.* **2022**, *31*, 101721. [CrossRef]
33. Hechler, E.; Oberhofer, M.; Schaeck, T. AI Information Architecture. *Deploying Enterp.* **2020**, 55–93. [CrossRef]
34. Fountaine, T.; Brian Saleh, T.M.; Brian Saleh, T.M. Building the AI-Powered Organization. *Harv. Bus. Rev.* **2019**, *97*, 62–73. [CrossRef]
35. Fisman, D.; Rosu, G. Tools and Algorithms for the Construction and Analysis of Systems. In Proceedings of the 28th International Conference, TACAS 2022, Munich, Germany, 2–7 April 2022; p. 13243.
36. Brühl, V. Agile methods in the German banking sector: Some evidence on expectations, experiences and success factors. *J. Bus. Econ.* **2022**, *92*, 1337–1372. [CrossRef]