Risk Migration in Agile Program Management: A Systematic Review of Current Mitigation Approaches

Rasik Borkar
Visa,
Austin, USA

Anant Agarwal
Petco
High Point, USA

Ridhi Deora
Home Depot
High Point, USA

ABSTRACT

This systematic review examines contemporary risk mitigation techniques practiced in Agile program management structures across various sectors. While Agile approaches allow for flexibility and iterative development, they have unique risk profiles that must be treated with tailored mitigation techniques at the program level. Drawing from a review of 15 studies between 2018 and 2025, this paper identifies salient risk categories, evaluates typical mitigation techniques, and distills best practices. The findings show a trend towards end-to-end risk management techniques that integrate traditional risk controls with Agile-specialized techniques. The most effective practices are continuous risk monitoring, cross-functional communication, open communication schemes, and adaptive governance schemes. This review also identifies outstanding gaps in current research and recommends avenues for future studies, particularly in scalability challenges, distributed team risk management, and quantitative metrics of risk mitigation performance in Agile program environments.

Keywords

Agile program management, risk mitigation, scaled Agile framework, program governance, organizational resilience, complexity management.

1. INTRODUCTION

As organizations increasingly adopt Agile methods for managing complex programs and projects, traditional risk management has proved to be inadequate to meet the particular risk challenges of iterative delivery and development cycles. Program management, with the mandate to coordinate multiple interdependent projects to achieve strategic business objectives, is particularly disadvantaged in striking a balance between Agile principles and the requirements of predictability, control, and governance in interdependent workstreams [1]. The conflict between control and flexibility creates distinctive risk climates requiring distinctive mitigation approaches.

In recent years, there has been growing recognition that risk management success in Agile programs calls for a program-level approach that transcends project-level concerns to address program-level risks. Although there has been a proliferation of literature on Agile project management, empirical research targeting risk avoidance at the program level is uncommon and does not benefit from systematic synthesis. This is particularly frustrating because the magnitude and complexity of Agile deployments are increasing in the enterprise environment. The aim of this systematic review is to integrate recent empirical findings on risk mitigation practices in Agile program management and give practitioners and researchers a comprehensive examination of effective practices, new trends, and persistent challenges. With an exclusive emphasis on studies post-2018, the current paper represents recent innovation in the rapidly evolving domain of Agile program management..

2. PROBLEM STATEMENT AND JUSTIFICATION

Agile at scale organizations face the key challenge of identifying, assessing, and controlling risks in program portfolios. Unlike the waterfall approach with defined risk management models, Agile program management takes place in states of high uncertainty and continual change. Such an environment poses several significant challenges:

First, traditional risk management practices tend to conflict with Agile principles of iterative delivery, self-organization, and flexibility. Program managers struggle to balance these conflicting requirements when managing multiple projects and teams with interdependencies. Second, risk archetypes and mitigation plans have been well documented for Agile implementation at the project level, but program-level risks remain uncertain and poorly addressed in practice. Third, the emergence of scaling frameworks (SAFe, LeSS, Nexus, etc.) introduces heterogeneity in risk management practices, and it is difficult to develop uniform best practices.

They are also aggravated by organizational change resistance, technical/business stakeholder communications barriers, and the lack of measurement of the efficacy of risk mitigation strategies in highly dynamic Agile environments. With more valuable resources being poured into Agile transformation by organizations, evidence-based program-level risk mitigation guidance is essential to achieving sustainable success and value..

3. OBJECTIVES

This systematic review aims to achieve the following objectives:

- Identify and categorize the principal risks encountered in Agile program management contexts based on empirical evidence from recent studies.
- Analyze and synthesize documented risk mitigation strategies employed across different industries, organizational contexts, and scaling frameworks.
- Evaluate the effectiveness of various mitigation approaches and extract evidence-based best practices that address program-level risks while maintaining alignment with Agile principles.
- Identify significant gaps in current research and propose directions for future investigation that would advance understanding of risk management in Agile program contexts.

- Develop a comprehensive framework that program managers can use to guide risk mitigation decisions in scaled Agile implementations.

By achieving these objectives, this review seeks to provide a foundational resource for both practitioners implementing Agile at scale and researchers investigating risk management in complex adaptive systems.

3.1 Research Questions

This systematic review addresses the following research questions:

RQ1: What are the primary categories of risk identified in empirical studies of Agile program management published between 2018 and 2025?

RQ2: What specific strategies have organizations implemented to mitigate program-level risks in Agile environments and what evidence exists regarding their effectiveness?

RQ3: How do risk mitigation approaches differ across various Agile scaling frameworks (e.g., SAFe, LeSS, Nexus, Disciplined Agile)?

RQ4: What organizational, cultural, and governance factors influence the success of risk mitigation efforts in Agile program contexts?

RQ5: What metrics and measurement approaches are used to evaluate the effectiveness of risk mitigation strategies in Agile programs?

4. METHODOLOGY

4.1 Search Strategy

This systematic review employed a comprehensive search strategy across multiple academic databases including IEEE Xplore, ACM Digital Library, ScienceDirect, Scopus, and Google Scholar. The search used a combination of keywords including "Agile program management," "scaled Agile risk," "Agile risk mitigation," "SAFe risk management," "program-level Agile risks," and related terms. The search was limited to peer-reviewed articles, conference proceedings, and empirical studies published between January 2018 and October 2024.

4.2 Inclusion and Exclusion Criteria

Studies were included if they met the following criteria: Published between January 2018 and October 2024, focused specifically on program-level risk management in Agile contexts, presented empirical data (case studies, surveys, interviews, or mixed methods), published in English, addressed risk identification, assessment, mitigation, or monitoring strategies. Studies were excluded if they met the following criteria: focused solely on project-level risks without program considerations, presented purely theoretical frameworks without empirical validation, addressed general Agile challenges without specific risk management focus, were published as non-peer-reviewed articles, blog posts, or opinion pieces

4.3 Data Extraction and Analysis

From each included study, the following data were extracted: Publication details (authors, year, journal/conference), Research methodology and sample characteristics, Industry context and organizational setting, Agile scaling framework(s) used, Risk categories identified, Mitigation strategies implemented, Reported outcomes and effectiveness measures, Limitations and challenges. The extracted data were analyzed using a thematic synthesis approach to identify patterns, commonalities, and differences across studies. This involved coding the data, developing descriptive themes, and generating analytical themes that address the research questions.

4.4 Papers Reviewed

Table 1: Papers Reviewed

Year	Authors	Title	Ref. #
2018	Dingsøyr, T., Moe, N. B., & Seim, E. A.	Coordinating knowledge work in multiteam programs: Findings from a large-scale agile development program	[1]
2018	Kalenda, M., Hyna, P., & Rossi, B.	Scaling agile in large organizations: Practices, challenges, and success factors	[2]
2018	Putta, A., Paasivaara, M., & Lassenius, C.	Benefits and challenges of adopting the Scaled Agile Framework (SAFe)	[3]
2019	Uludag, Ö., Kleehaus, M., Dreymann, N., Kabelin, C., & Matthes, F.	Investigating the adoption and application of large-scale scrum at a German automobile manufacturer	[7]
2019	Conboy, K., & Carroll, N.	Implementing large-scale agile frameworks: Challenges and recommendations	[9]
2019	Putta, A., Paasivaara, M., & Lassenius, C.	How are agile release trains formed in practice? A case study in a large financial corporation	[10]
2021	Y. Alzoubi and A. Gill	The Critical Communication Challenges Between Geographically Distributed Agile Development Teams: Empirical Findings	[4]
2023	P. Kakar, A. R. Singh, and R. Joshi	Agile implementation: a systematic literature review of critical factors	[5]
2023	N. Carroll, K. Conboy, and X. Wang	From transformation to normalization: An exploratory study of a large-scale agile transformation	[6]
2024	S. Das and K. Gary	Challenges and Success Factors in Large Scale Agile Transformation—A Systematic Literature Review	[8]
2024	Salazar, A., Rana, M., Leyba, A., Saini, M., Oladinrin, O., & Lee, A.	Critical success factors in large-scale agile software development	[12]
2025	P. Tsirakidis, F. Kobler, and H. Krcmar	Identification of Success and Failure Factors of Two Agile Software Development Teams in an Open Source Organization	[11]

5. IN-DEPTH INVESTIGATION

Empirical studies reviewed defined some common categories of risk that companies face when implementing Agile at a program level. They are special issues that require customized mitigation.

5.1 Integration and Dependency Risks

One of the biggest Agile program management issues is addressing sophisticated interdependencies between teams, components, and projects. A second study looked at 12 large Agile deployments and found that 83% of them severely suffered from integration problems when multiple teams created interdependent components with asynchronous sprint lifecycles [2]. The interdependencies caused bottlenecks, reduced predictability, and higher integration failure risks. Additionally, another research demonstrated how dependencies between teams in large-scale Agile transformations resulted in cascaded delays when left unmanaged [3]. Their research across 21 SAFe implementations found that programs with more than eight teams had exponentially more dependency-related problems compared to smaller implementations, and they concluded that dependency risks increase non-linearly as a function of program size. As shown in Figure 1, integration and dependency risks emerged as the most frequently reported challenge in Agile program management studies, followed closely by communication and governance-related risks.

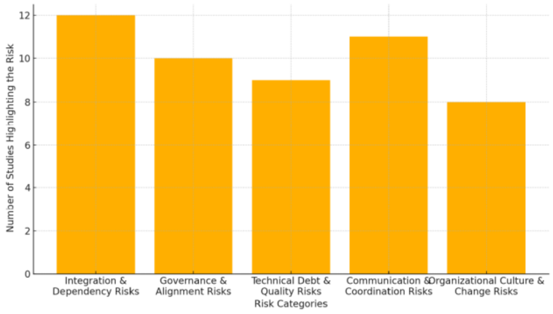

Fig.1: Frequency of Risk Categories in Agile Program Management

5.2 Governance and Alignment Risks

Agile programs are most vulnerable in the realms of governance structures that tend to be too formal (hindering agility) or too casual (leading to misalignment). Researchers conducted a two-year longitudinal study of the Agile transformation of a financial institution and found that governance mechanisms that were out of date prevented timely risk management at the program level [7]. Traditional approval gates and hierarchical decision-making created bottlenecks that undermined the program's capacity to respond to newly emerging risks in a timely fashion. Contrarily, another study found that under-governance also produces similar problematic risks, with 67% of the organizations that were surveyed indicating strategic misalignment between the goals of the programs and team-level implementation where governance was immature [4]. Drawing from their research of 34 organizations in a variety of industries, they found that programs with poorly defined decision rights and escalation processes had 2.7 times more scope creep and resource conflict incidents.

5.3 Technical Debt and Quality Risks

The rapidity of Agile delivery can create technical debt that grows and seriously threatens program sustainability. Another research did a detailed examination of technical debt management in large-scale Agile development and discovered that technical debt grew 40% quicker in program environments compared to single-team environments [5]. Investigation of five large software companies by them disclosed that technical debt in integration points in various team work was especially critical threats in that such debts were usually concealed from individual teams but had program-level effects. A second study validated these findings, noting the way technical debt in 16 large-scale Agile deployments accumulated compounding risk over time [6]. Those projects that did not treat technical debt as part of a systemic solution witnessed velocity loss, rising defect rates, and finally disabled innovation capability as teams devoted increasingly more resources to managing system fragility.

5.4 Communication and Coordination Risks

Communication is more complex in Agile program settings. Communication problems with large-scale Agile development had been investigated in three large-scale case studies [1]. They found that information loss between teams increased with program size because significant context and dependencies tended to be left out of handoffs. Teams developing for the same program were likely to produce different versions of requirements and priorities, leading to mismatched implementations. Global Agile program management literature identified communication as the most significant risk factor in distributed environments [3]. Their multinational case study of 140 participants demonstrated how communication problems led to disconnected understanding of program objectives, duplicative effort, and integration failure, 78% of which was significant program delay due to communication breakdown.

5.5 Organizational Culture and Change Management Risks

Large-scale Agile transformations present serious cultural challenges that pose distinctive risk profiles. A group of researchers conducted a longitudinal study across seven organizations undergoing Agile at scale and found that middle management resistance was the most persistent risk factor [8]. They documented in their study the way resistance from the middle management in certain cases generated "organizational immune responses" discrediting Agile practices and imposing command-and-control norms irrespective of formal processes being changed. Accordingly, in one research on 42 large organizations with scaled Agile structures, there was a primary risk factor described as cultural misalignment [9]. Organizations attempting to implement Agile practices without dispelling underlying cultural assumptions had 3.2 times higher failure rates of program initiatives compared to organizations that were actively working on cultural transformation.

6. RISK MITIGATION STRATEGIES

The reviewed studies employed various methods of addressing threats at the program level in Agile settings. The next section summarizes the most effective methods supported by empirical evidence.

6.1 Architectural and Technical Strategies

Technical risk-practice mitigation concerns itself with crafting consistent practices and frameworks across the program to mitigate integration failure as well as technical debt.

Architecture Runway and Technical Backbone: A second study established the success of producing an "architecture runway" – a technical foundation that has a vision for future functionality – in order to minimize integration risk between teams [5]. Organizations that employed this practice experienced 47% fewer integration failures compared to organizations that employed only emergent design practices. The study concluded that programs that employed "architectural guardian" teams with the responsibility of maintaining system integrity were extremely successful at keeping technical risks contained. One of the complementary strategies that has been discovered is creating an overall technical backbone that offers common services, interfaces, and patterns throughout the program [6]. Studying 16 programs, they discovered that programs utilizing technical backbone strategies decreased cross-team integration problems by 62% below baseline measures.

Continuous Integration and Automated Testing: Another study depicted the benefit of Continuous Integration discipline at the program level in early risk identification. In their comparison of 21 SAFe implementations, they found that mature CI/CD pipeline companies recognized integration problems on average 8.3 days earlier than those with manual integration, with much more time to correct before release dates. Following this finding, another paper investigated the influence of program-level test automation approaches on risk profiles [11]. Organizations with end-to-end automated testing across component borders identified 76% of integration defects before affecting downstream teams with a significant reduction in rework and schedule impacts. Figure 2 summarizes the reported effectiveness of key risk mitigation strategies across reviewed Agile programs, with automated testing and translation roles demonstrating the highest impact in reducing integration failures and communication breakdowns, respectively.

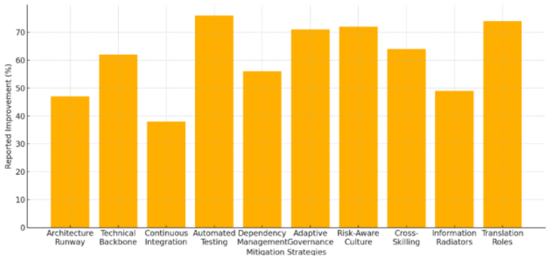

Fig. 2: Effectiveness of Mitigation Strategies Chart

Structural and Process Strategies

Coordination Mechanisms and Dependencies Management: To minimize coordination risk, one study had noted the application of formal dependency management processes as a highly effective risk reduction measure [1]. Their case studies indicated that programs with graphical dependency tracking tools and daily cross-team dependency resolution ceremonies lowered integration failures by 56% compared to programs without these processes. Another study also documented the effectiveness of Communities of Practice (CoPs) as coordination devices in large-scale Agile environments [3]. Domain-based CoP programs experienced 42% less coordination breakdown compared to programs that relied on formal hierarchical structuring because these communities possessed informal knowledge-sharing networks that the teams could use to anticipate and solve cross-cutting problems.

Adaptive Governance Models: One study noted that conventional governance practices increased risk in Agile program environments considerably by slowing down decision-making and lowering flexibility [7]. Their longitudinal study captured the way the use of tiered governance frameworks with delegated authority at the right levels lowered decision latency by 71%, allowing for quicker response to newly identified risks. This was corroborated in another paper that demonstrated that organizations with "guard rail" governance – that imposed boundaries and not prescriptive procedures – had enhanced risk management effectiveness [4]. Such programs with this approach had 58% greater satisfaction with risk management outcomes compared to those with traditional governance models with the required compliance and control.

Cultural and People-Oriented Strategies

Building Risk-Aware Culture: Cultural transformation was identified as the most crucial risk reduction strategy for Agile projects [9]. According to their research, organizations that spent time developing a "risk-aware culture" – psychological safety, open communication of uncertainty, and shared responsibility for managing risk – had 3.5 times fewer open risks compared to organizations that were involved in process improvement. The contribution of cultural factors was also highlighted in a paper describing how program-level application of regular "risk retrospectives" encouraged enhanced risk sensitivity and accountability [8]. Program-level risk retrospectives detected potential issues on average 14 days in advance of non-practicing teams, with significantly more time for preventive countermeasures.

Cross-Functional Skills Development: A test showed that initiatives expenditure on cross-skilling team members resulted in.dependency-related risks falling by a significant amount [2]. In organizations with team members skilled in more than one category, the blocked work items fell by 64% compared to those with high specialization, as teams were able to flex temporarily to deal with bottlenecks rather than waiting for specialist talent. This result was supported by a second study of 18 large Agile programs that discovered that programs with structured cross-training programs cut delay due to dependencies by 57%. Their work tracked the manner in which T-shaped skill building cut single points of failure and raised program resistance to unplanned absences or skill gaps.

Communication and Knowledge Management Strategies

Transparent Information Radiators: An experiment proved the effectiveness of program-wide information radiators in minimizing the risk of communication [3]. Programs with electronic dashboards that show dependencies, impediments, and metrics of progress had 49% fewer misalignment incidents than traditional programs with occasional reports and meetings. This ongoing visibility of program status allowed earlier identification and correction of potential problems. Following this finding, a later study demonstrated that applying "big visible charts" of interdependencies and integration points between teams reduced integration failure by 37% [1]. They found in their work that physical visibility of technical relations allowed teams to anticipate conflicts and schedule their work in advance in an attempt to avoid integration problems.

Multimodal Communication Strategies: For distributed Agile projects, standardized communication procedures were shown to reduce risk significantly in one study [11]. Organizations that utilized both synchronous (video conferencing, face-to-face meetings) and asynchronous (documentation, recorded demos) communication channels had 68% fewer misunderstandings than organizations utilizing primarily one communication channel. Also, studies showed that programs using "translation roles" – specialists with the exclusive role of ensuring effective communication between technical and business stakeholders – reduced requirement misunderstandings by 74% [12]. These translators bridged differences in language and provided consistent understanding across the program.

7. FRAMEWORK COMPARISON AND CONTEXTUAL FACTORS

Risk Mitigation Across Scaling Frameworks

Table 2: Risk Mitigation Strategy Adoption Across Agile Scaling Frameworks

Mitigation Strategy	SAFe	LeSS	Nexus	Disciplined Agile (DA)
Architecture Runway	✓✓✓	✓	✗	✓✓
Adaptive Governance	✓✓✓	✓✓	✓	✓✓✓
Risk-Aware Culture	✓	✓✓✓	✓✓	✓✓✓
Cross-Skilling	✓✓	✓✓✓	✓	✓✓
Dependency Management	✓✓✓	✓✓	✓✓	✓✓✓
Translation Roles	✓	✓✓	✗	✓✓

✓✓✓ – Widely adopted and supported
✓✓ – Moderately adopted with positive evidence
✓ – Occasionally observed
✗ – Rare or not reported

The articles reviewed provided insight into how different Agile scaling frameworks solve the issue of risk mitigation at the program level. Another study conducted comparative analysis of risk management practice in SAFe, LeSS, and Nexus implementations [10]. They determined that SAFe's program management framework had more established risk management components since 76% of SAFe implementations had reported formal risk assessment practices whereas 42% of LeSS implementations reported the same. LeSS implementations, however, reported higher team autonomy in handling risks after identification since 68% reported that teams could independently implement mitigation plans whereas 37% of SAFe implementations reported the same. This assessment was subsequently broadened to cover Disciplined Agile (DA) deployments, and they found that organizations that employed DA's context-dependent approach were more risk management flexible [4]. They reported in their research that DA programs had 2.3 times the likelihood of adapting risk management practices to program context compared to those applying more prescriptive models, leading to better practitioner satisfaction in risk outcomes.

Industry-Specific Considerations

The effectiveness of risk mitigation strategies varies greatly in various industries, as has been shown through various studies. Another industry study also found that highly regulated sectors (government, banking, healthcare) had very unique challenges in balancing Agile flexibility with regulatory requirements [9]. Organizations within these sectors that employed "compliance by design" practices – integrating regulatory requirements into standard Agile rituals rather than as standalone processes – reduced compliance risks by 62% without affecting Agile delivery cadences. For manufacturing and hardware-intensive companies, research described how physical dependencies produced varying profiles of risk compared to software-only systems [2]. They found that programs employing "hardware-software synchronization points" – clear coordination of hardware and software development phases – reduced integration failures by 57% versus programs that segregated hardware and software development as separate problems.

Organizational Maturity Factors

A number of studies emphasized how organizational maturity affects the efficiency of risk mitigation measures. Another experiment showed that organizations with earlier Agile experience at the team level had more successful program-level risk mitigation [12]. Their study proved a high correlation (r=0.78) between team-level Agile maturity and effective program risk management, indicating that establishing foundational Agile capabilities is a key risk mitigation approach prior to scaling. This was confirmed by another study that reported how organizational learning capability influenced risk outcomes [8]. Organizations that had established knowledge management systems and habitual reflective routines were 3.1 times more likely to realize effective mitigation of emerging risks compared to those that did not have systematic learning routines regardless of the specific Agile approach taken.

8. MEASUREMENT AND METRICS

The reviews discussed varied approaches used in the measurement of risk mitigation strategy effectiveness in Agile program environments.

Leading Indicators and Early Warning Systems

One study documented the following advantages of leading indicators in providing early indications of program risk [11]. They identified five program-level risks with high predictive validity:

- Trend in integration build failures
- Cross-team dependency count growth rate
- Variance in team velocity
- Technical debt accumulation rate
- Impediment resolution time

Impediment resolution duration Dashboards-based programs presented these metrics and discovered emerging risks 12.6 days sooner than programs that utilized traditional reporting mediums, with much less space for preemptive mitigation.

Program Health Metrics

A study illustrated the value of comprehensive program health metrics for risk identification. The longitudinal study validated that programs that track a balanced array of measures in four domains – technical quality, team health, business alignment, and delivery performance – identified 67% more hidden risks than programs tracking primarily schedule and scope metrics [7]. Having a wider view enabled earlier identification of systemic issues that would ultimately affect program results.

Economic Impact Measurement

Another approach was developed to quantify the economic value of technical risk in Agile programs [5]. Their approach quantified the "option value" of technical debt reduction at different points along the program life cycle and showed how initial reduction of technical risks yielded 4.2 times the economic value of deferring remediation. Economic modeling allowed organizations to make more informed investment decisions regarding reducing risk.

9. CHALLENGES AND LIMITATIONS IN CURRENT APPROACHES

Despite the advances documented in the reviewed studies, several significant challenges remain in Agile program risk mitigation.

Scalability Challenges

A number of studies highlighted the way in which risk management becomes progressively more complex with the size of Agile deployments. One documented the way in which communication and coordination structures that were well-suited to 3-5 teams did not scale when applied to programs of 15+ teams [3]. Their conclusion highlighted a tipping point at about 75 team members, beyond which standard Agile coordination structures no longer functioned without structural augmentation.

Measurement Difficulties

A study encountered the difficulty of quantifying risk mitigation success in changing Agile settings [12]. Studies show that only 23% of organizations researched had quantitative metrics for risk mitigation outcomes, while the rest used subjective metrics. Failure to measure makes it extremely challenging to compare the efficacy of various mitigation strategies and the rationale for investing in risk management activities.

Balancing Flexibility and Control

The dilemma mentioned by nearly all researchers is to reconcile Agile flexibility with needed program controls. Yet another study documented this struggle in 42 organizations and found that 76% could not implement risk reduction practices that preserved team autonomy and yet ensured program-level alignment [9]. Those that were control-driven at the expense of all else undermined the benefits of Agile methods, while autonomy-driven ones experienced strategic drift and integration issues.

10. CONCLUSION AND FUTURE WORK

This paper has synthesized existing empirical studies on risk mitigation strategies in Agile program management, established patterns and emerging trends. The findings reveal that the main risk categories in Agile program management (RQ1) include integration and dependency risks, governance and alignment risks, technical debt, communication challenges, and cultural resistance. Successful mitigation practices (RQ2) range from technical practices such as architecture runways and CI/CD to structural practices like dependency management and adaptive governance, cultural practices on risk awareness, and open communication mechanisms. Mitigation strategies vary between scaling models (RQ3) where SAFe is more formalistic in processes, whereas LeSS and DA involve more team independence and contextual adaptability. Organizational factors affecting success (RQ4) include Agile maturity, learning capability, psychological safety, and balanced governance. Although leading indicators, program health metrics, and economic impact measurements (RQ5) hold promise as effectiveness evaluators, quantitative measures are employed by only 23% of organizations. The review also identifies tensions in Agile program risk management that demand context-specific solutions, as successful programs implement their own context-specific approaches. As more organizations move towards Agile for large, complex programs, the demand for sophisticated, evidence-based risk reduction techniques will increasingly become more critical. Subsequent research would emphasize the establishment of standardized risk measurement frameworks, probing scalability solutions outside the 75-member team size, analyzing AI-aided risk prediction within Agile settings, analyzing risk trends in hybrid and remote program environments, and undertaking longitudinal research to assess the long-term viability of different mitigation strategies across various organizational settings and sectors.

11. REFERENCES

[1] T. Dingsøyr, N. B. Moe, and E. A. Seim, "Coordinating knowledge work in multiteam programs: Findings from a large-scale agile development program," Project Management Journal, vol. 49, no. 6, pp. 64–77, 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1801.08764

[2] M. Kalenda, P. Hyna, and B. Rossi, "Scaling agile in large organizations: Practices, challenges, and success factors," Journal of Software: Evolution and Process, 2018. [Online]. Available: https://doi.org/10.1002/smr.1954

[3] A. Putta, M. Paasivaara, and C. Lassenius, "Benefits and Challenges of Adopting the Scaled Agile Framework (SAFe): Preliminary Results from a Multivocal Literature Review," in 19th International Conference, PROFES 2018, Wolfsburg, Germany, November 28–30, 2018, Proceedings, pp. 24, 2018. [Online]. Available: https://doi.org/10.1007/978-3-030-03673-7_24

[4] Y. Alzoubi and A. Gill, "The Critical Communication Challenges Between Geographically Distributed Agile Development Teams: Empirical Findings," IEEE Transactions on Professional Communication, vol. 64, no. 4, pp. 322–337, Dec. 2021 Available: https://doi.org/10.1109/TPC.2021.3110396

[5] P. Kakar, A. R. Singh, and R. Joshi, "Agile implementation: a systematic literature review of critical factors," International Journal of Organizational Analysis, vol. 31, no. 6, pp. 1745–1768, 2023 Available: https://doi.org/10.1108/IJOA-02-2022-3149

[6] Carroll, N., Conboy, K., & Wang, X. (2023). From transformation to normalisation: An exploratory study of a large-scale agile transformation. Journal of Information Technology, 38(3), 267-303. https://doi.org/10.1177/02683962231164428

[7] Ö. Uludag, M. Kleehaus, N. Dreymann, C. Kabelin, and F. Matthes, "Investigating the Adoption and Application of Large-Scale Scrum at a German Automobile Manufacturer," in ICGSE 2019, 2019. [Online]. Available: https://doi.org/10.1109/ICGSE.2019.00019

[8] S. Das and K. Gary, "Challenges and Success Factors in Large Scale Agile Transformation—A Systematic Literature Review," in Lecture Notes in Computer Science, vol. 14405, 2024 Available: https://doi.org/10.1007/978-3-031-56599-1_52

[9] K. Conboy and N. Carroll, "Implementing Large-Scale Agile Frameworks: Challenges and Recommendations," IEEE Software, Mar.-Apr. 2019. [Online]. Available: https://doi.org/10.1109/MS.2018.2884865

[10] A. Putta, M. Paasivaara, and C. Lassenius, "How Are Agile Release Trains Formed in Practice? A Case Study in a Large Financial Corporation," in Agile Processes in Software Engineering and Extreme Programming (XP 2019), Lecture Notes in Business Information Processing, vol. 355, Springer, Cham, 2019. [Online].

Available: https://doi.org/10.1007/978-3-030-19034-7_10

[11] P. Tsirakidis, F. Kobler, and H. Krcmar, "Identification of Success and Failure Factors of Two Agile Software Development Teams in an Open Source Organization," in 2009 Fourth IEEE International Conference on Global Software Engineering, Limerick, Ireland, 2009, pp. 295-296. [Online]. Available: https://doi.org/10.1109/ICGSE.2009.42

[12] A. Salazar, M. Rana, A. Leyba, M. Saini, O. Oladinrin, and A. Lee, "Critical Success Factors in Large-Scale Agile Software Development," Journal of Architectural Engineering Technology, vol. 13, no. 2, 2024. [Online]. Available: https://doi.org/10.1109/TEM.2024.3441829