

# Approaches to Managing Large Development Teams in High-load Projects

Garg Nishant\*

*Software Development Manager, Amazon, Seattle, Washington, USA*

*Email: nishgarg@amazon.com*

## Abstract

This article explores approaches to managing large teams under high-load conditions using Agile methodologies. Based on a systematic analysis of publicly available literature, the historical background and evolution of agile methodologies were examined, and key challenges related to organizational, communication, and technical aspects of scaling Agile in large distributed structures were identified. Special attention is given to practical strategies such as forming a unified high-level backlog, fostering a culture of collaboration, supporting training and certification, reducing iteration length, utilizing modern digital tools, and prioritizing testing. The findings demonstrate that the comprehensive application of these strategies contributes to reducing development time, improving product quality, and optimizing resource management under high-load conditions. The insights presented in this article will be valuable to technical directors, engineering department managers, program and product team leaders, as well as other researchers in organizational design and DevOps practices interested in developing and implementing effective approaches to managing large distributed teams within high-load and scalable digital ecosystems.

**Keywords:** agile methodologies; Agile scaling; large teams; high load; organizational challenges; digital tools; project management.

## 1. Introduction

In the modern, rapidly evolving business environment and the fast-paced development of information technologies, companies must adapt their processes to respond swiftly to market changes. Managing large development teams in high-load projects presents a particular challenge, as traditional management methods often lack the necessary flexibility.

---

*Received:* 6/25/2025

*Accepted:* 8/9/2025

*Published:* 8/19/2025

---

\* Corresponding author.

Agile methodologies, which have proven effective in small and medium-sized projects, offer significant advantages in dealing with uncertainty and rapidly changing requirements. However, adapting these approaches for large, distributed structures remains problematic, highlighting the need to explore Agile scaling strategies for managing large teams under high-load conditions.

An analysis of research on managing large development teams in high-load projects identifies three thematic areas, each contributing to a comprehensive understanding of Agile approaches.

The first area includes theoretical reviews and strategic studies focused on scaling Agile methodologies. The online resource *Scaling Agile: Strategies for Managing Large Development Teams* [1] presents practical strategies for adapting Agile methodologies to large teams, while a review by Daraojimba E. C. and his colleagues [2] systematizes the experience of applying Agile approaches in project management, emphasizing their potential in high-load systems with dynamic requirements. A strategic aspect is also reflected in the study by Alves E. J. and Gonçalves C. A. [4], where Agile portfolio management is considered a key factor in maintaining market competitiveness. These insights support the hypothesis that integrating Agile methodologies with traditional management strategies can enhance the responsiveness and adaptability of large development teams. Collectively, these studies provide a crucial strategic vision, but they often remain at a high level of abstraction, lacking detailed guidance on how to resolve the ground-level organizational and communication conflicts that arise in large, globally distributed teams.

The second area covers methodological developments aimed at practical testing and knowledge transfer related to Agile approaches. A study conducted by Berbegal-Mirabent J., Gil-Doménech D., and Berbegal-Mirabent N. [3] explores the integration of Agile methods into project management education, not only fostering practical skills but also encouraging analytical thinking. In parallel, the development of a decision-making algorithm for selecting an optimal set of Agile tools, proposed by Andreev V. V. and Malozemov S. N. [5], provides a methodological foundation for informed managerial decisions in engineering and industrial projects. While valuable for developing skills and making tool-related decisions, these works tend to treat Agile adoption as a procedural or technical challenge, often understating the profound cultural and organizational shifts required for these methodologies to succeed at scale.

The third area comprises empirical and comparative studies assessing the impact of Agile approaches on project management effectiveness in real-world scenarios. A study by Iqbal J. and his colleagues [6] demonstrates the positive influence of Agile methodologies on development processes in Pakistani IT companies, supporting the hypothesis that Agile enhances project execution speed and quality. Similar empirical findings are presented in the work of Pal K. and Karakostas B. [7], which examines software testing practices in Agile, Scrum, and DevOps environments. However, a comparative analysis by Halani K. R. and Jhajharia K. [8], along with studies on the choice between Scrum and Kanban by Zasornova I., Lysenko S., and Zasornov O. [9], reveal conflicting results when comparing Agile approaches with the traditional Waterfall model. This indicates the need for a deeper analysis of factors influencing the adaptation of Agile methodologies in different organizational and cultural contexts. These studies employ quantitative analysis, case studies, and comparative approaches, providing a comprehensive examination of the issue but leaving a research gap in the systematic

classification of criteria for selecting an optimal Agile methodology for high-load projects.

Review of existing literature shows that the majority of research is dedicated to the application of Agile in small and medium-sized teams. Collectively, the available literature provides a fragmentary but deep understanding of agile scaling. Studies are divided into strategic reviews, specific methodological tools, and individual empirical investigations. Meanwhile, structured knowledge and practical recommendations for implementing Agile methodologies to manage large developer teams under high-load conditions remain limited. There is no single integrated framework that bridges the gap between high-level strategy and on-the-ground practice, addressing organizational, process-related, and cultural challenges in a unified manner. Such fragmentation creates a significant gap in research on an integrated model that could assist organizations in their complex process of agile scaling in high-load, multi-team environments. The absence of a comprehensive approach to agile scaling necessitates further empirical and theoretical research aimed at identifying key success factors in large organizations.

The objective of this study is to systematize the theoretical foundations and practical approaches to managing large development teams in high-load projects using Agile methodologies.

The scientific novelty lies in the comprehensive analysis of existing Agile frameworks, considering the specifics of scalable projects and high-load environments, as well as the development of an integrated management model that combines traditional and Agile approaches. It is assumed that the comprehensive application of these strategies will lead to a reduction in time-to-market and an improvement in the execution quality of high-load projects. However, the research is not without limitations, as the methodology is limited to a systematic analysis of existing publicly available literature and does not involve the collection of new primary data through surveys, interviews, or case studies. In addition, although the strategies discussed are intended to be widely applied, this study recognizes that their practical effectiveness largely depends on specific conditions, such as corporate culture, team maturity, and industry specifics. To achieve this objective, the study employs a systematic analysis of the literature.

## **2. Theoretical foundations and evolution of agile methodologies in managing large teams**

Modern project management methodologies have undergone significant changes, transitioning from traditional models such as Waterfall to agile approaches. Initially emerging in the software development industry in the early 2000s, Agile methodologies were developed in response to the shortcomings of classical models, which struggled to adapt to rapidly changing customer requirements and market dynamics [8]. A key milestone in the evolution of agile approaches was the publication of the Agile Manifesto in 2001, which established four core values and twelve principles emphasizing iterative development, continuous client interaction, and adaptability to change. The key principles of Agile include iterative planning, continuous feedback, cross-functional team collaboration, and self-organization. These concepts provided the foundation for dynamic project management and improved adaptability in uncertain conditions [3]. Over time, various specialized frameworks such as Scrum, Kanban, Lean, and Extreme Programming (XP) emerged, each with distinct characteristics and applications. For instance, Scrum focuses on working in short iterations (sprints) with well-defined roles and

events, while Kanban emphasizes workflow optimization through task visualization and work-in-progress (WIP) limitations [9]. The transition from development within small, self-organized teams to scaling Agile in large organizations necessitates a reassessment of fundamental principles. In large and distributed teams, coordination, synchronization of priorities, and a unified product vision become critical, requiring the integration of Agile approaches with traditional management tools [2]. Thus, the evolution of Agile methodologies reflects not only technical and methodological innovations but also the transformation of organizational culture aimed at enhancing flexibility and responsiveness under high-load conditions. When scaling Agile methodologies to large-scale projects, particular attention is given to adaptation and implementation strategies. In large organizations with over 150–200 employees, traditional Agile frameworks require additional coordination mechanisms, such as forming a unified high-level backlog, inter-team synchronization of iterations, and organizing regular knowledge-sharing meetings [6]. Scaled Agile frameworks such as SAgile (Scaled Agile Framework), LeSS (Large-Scale Scrum), and Nexus have been specifically developed to address these challenges, enabling the integration of agile methodologies into enterprise-level project management systems [1]. Furthermore, successful Agile scaling depends on fostering a culture of collaboration, continuous employee training, and leveraging specialized digital tools (e.g., Wrike, Trello) that enhance process automation and communication transparency [2, 5]. Collectively, these factors create conditions for efficient management of large teams, ensuring high adaptability and responsiveness in the execution of high-load projects. For a deeper understanding, Table 1 provides a comparison of key Agile methodologies, considering their principles, main features, advantages, and limitations.

**Table 1:** Comparison of key Agile methodologies [1, 3, 9].

Methodology	Core Principles	Key Features	Advantages	Limitations
Scrum	Iterative planning, sprint-based work, defined roles (Scrum Master, Product Owner, Development Team)	Structured workflow, regular stand-up meetings, sprint reviews	High transparency, rapid adaptability, results-oriented approach	Requires strict role adherence, challenging to scale for large teams
Kanban	Continuous flow, work visualization, work-in-progress (WIP) limits	Flexible adjustments, no fixed iterations	Easy adaptation, improved workflow visibility, reduced delays	Less predictable timelines, lacks structured iterations
Lean	Waste minimization, process optimization, continuous improvement	Focus on customer value, elimination of redundant actions	Increased efficiency, cost reduction, improved quality	Difficult to integrate into traditional structures, requires deep process analytics
Extreme Programming (XP)	Pair programming, continuous testing, frequent integration, collective responsibility	High team interaction, automated testing emphasis	Enhanced code quality, fast feedback loops, defect reduction	Highly dependent on team expertise, challenging to scale for large projects

The evolution of Agile methodologies reflects the shift from highly specialized approaches for small teams to comprehensive management systems capable of addressing the needs of large and distributed teams. Recent studies confirm that effective scaling of agile methodologies requires not only the preservation of core principles but also their adaptation to the specific needs of large organizations through integration with traditional management methods and the use of modern digital tools.

### **3. Analysis of problems and challenges in managing large development teams under high-load conditions**

Managing large teams in high-load projects presents a range of challenges stemming from organizational and communication barriers, as well as technical and process-related constraints. Recent studies confirm that transitioning from Agile approaches, which are effectively applied in small groups, to scaling Agile across large distributed structures requires additional adaptation of flexibility principles and continuous improvement. The absence of a centralized mechanism for synchronizing priorities and the fragmentation of backlogs lead to inefficient resource allocation and reduced process transparency.

From a development process perspective, scaling Agile requires revising iteration synchronization methods, responsibility distribution, and module integration across multiple teams. The lack of well-defined coordination processes can result in delayed issue detection, prolonged release cycles, and a decline in product quality [1, 8]. Additionally, the implementation of modern digital tools for monitoring and project management often faces technical limitations, particularly when integrating data from different version control systems and test automation platforms.

Transitioning from traditional Agile practices designed for small teams to managing large-scale projects involves risks related to:

- Loss of iteration synchronization: As the number of teams increases, coordinating sprint planning and execution becomes more complex, potentially reducing development efficiency [6].
- Dilution of responsibility: Distributed accountability across multiple teams can lead to unclear role definitions and responsibility zones, negatively impacting product quality.
- Increased communication bottlenecks: Scaling Agile approaches introduces challenges in ensuring timely and effective communication among all project participants, requiring additional resources and tools [1, 3].

For a structured illustration of the main problems and challenges, Table 2 presents a comparative analysis.

**Table 2:** The main problems and challenges in managing large development teams [1, 3, 6].

Problem Category	Problem Description	Impact on Project	Recommended Solution Strategies
Organizational Barriers	Lack of coordination between distributed teams, fragmented project vision, differences in corporate culture	Delays, conflicts of interest, reduced motivation	Establishment of a unified high-level backlog, fostering a culture of collaboration, strengthening leadership
Communication Issues	Limited interaction between teams, language and time zone barriers, lack of transparency in communication	Implementation errors, delayed decision-making	Regular cross-team meetings, adoption of integrated digital communication platforms (e.g., Slack, Microsoft Teams)
Technical and Process Complexities	Challenges in synchronizing iterations, responsibility allocation, module integration, and test automation	Delayed releases, reduced product quality	Implementation of specialized project management tools (e.g., Wrike, Trello), standardization of processes, formation of cross-functional teams

A comprehensive analysis reveals that successfully managing large development teams under high-load conditions requires not only adapting Agile approaches but also making significant changes to the organizational structure. The key challenges include both internal (organizational and communication) and external (technical and process-related) factors, which collectively determine the project's success. Overcoming these challenges necessitates integrated strategies that combine traditional management methods with agile approaches, along with the implementation of modern digital tools that enhance automation and process transparency.

#### 4. Practical approaches and strategies for scaling agile practices in high-load projects

Managing large-scale projects under high-load conditions requires a focused adaptation of Agile approaches originally designed for small teams. Successful scaling of Agile methodologies must account for the complexities of distributed structures, variations in corporate culture, and the intricacies of technical processes. Recent studies highlight that applying a comprehensive set of practical strategies contributes to reducing time-to-market, improving development quality, and minimizing defect rates.

One of the key strategies for scaling Agile is the creation of a unified high-level backlog that consolidates the priority tasks of all teams. This approach ensures a shared vision of the final product and synchronizes the work of cross-functional groups. Clearly defined goals and objectives help eliminate misalignment between different departments, allowing for timely adjustments to the project plan [7].

Scaling Agile requires enhanced inter-team collaboration. Achieving a high level of cooperation necessitates fostering a corporate culture focused on continuous communication, knowledge sharing, and collective decision-making. Regular cross-team meetings, retrospective practices, and experience exchange contribute to the formation of a unified and consistent project approach [1,3].

In the context of Agile scaling, employee upskilling plays a crucial role. Organizations should invest in regular training, workshops, and certification programs aimed at deepening Agile methodology expertise. Well-trained teams adapt more easily to changes and implement innovative solutions, ultimately enhancing project efficiency [7]. A core element of effective Agile implementation is the use of short iterations, enabling rapid feedback loops and necessary adjustments. In high-load projects, shortened sprints help identify and resolve issues more quickly, reducing the risk of defect accumulation and ensuring stable development progress [5]. This approach also improves planning accuracy and resource management. Effective scaling of Agile practices is possible with the active adoption of specialized digital platforms for project management. Tools such as Wrike, Trello, Jira, and others facilitate the automation of planning, task tracking, team coordination, and feedback collection. Leveraging these tools enhances process transparency and accelerates decision-making [2,4]. To minimize risks and ensure high product quality, implementing prioritized testing practices is essential. Regular and automated testing enables the early detection of defects, reducing the likelihood of costly errors in later project stages. Thus, prioritized testing becomes an integral part of Agile scaling strategies, ensuring development transparency and stability (Table 3).

**Table 3:** Main strategies for scaling Agile practices [1, 6, 8, 9].

Strategy	Description	Key Benefits	Risks/Limitations
Backlog Alignment	Establishment of a unified high-level backlog consolidating priority tasks across teams	Synchronization of vision, reduction of misalignment risks	Requires clear role distribution and accountability
Fostering a Collaborative Culture	Creation of conditions for inter-team collaboration through regular meetings and retrospectives	Increased motivation, improved knowledge sharing	Potential communication barriers in distributed teams
Training and Certification Support	Implementation of training programs, certification initiatives, and experience-sharing sessions to enhance expertise	Higher professionalism, faster adaptation to changes	Requires continuous investment in education
Minimizing Iteration Length	Execution of short sprints to facilitate rapid feedback and immediate defect resolution	Faster adaptation, reduced defect accumulation	Potential increase in the number of iterations requiring tighter control
Use of Modern Digital Tools	Adoption of platforms for automating planning, task tracking, and communication	Enhanced process transparency, accelerated decision-making	Requires integration with existing systems and staff training
Prioritized Testing	Regular automated testing at all development stages	Reduced defect rates, improved final product quality	Dependent on automation level and team expertise

Strategies outlined in this study should be understood not as a simple menu of options but as an interconnected ecosystem. The comprehensive application of these approaches enables the creation of a flexible and scalable system for managing large developer teams under conditions of high workload. The combination of backlog alignment, active cultivation of corporate culture, continuous learning, optimization of iterative processes, integration of digital tools and prioritized testing contributes to the achievement of the project's strategic objectives — reduction of time-to-market, enhancement of development quality and optimization of resource management. Practical experience within large technology corporations confirms that the most serious challenges are seldom technical in nature: they arise from human factors such as misaligned priorities and gaps in communication among dozens of teams. Thus, a successful scaling initiative is one in which organisational alignment is prioritised over mere process optimisation. This synthesized framework, which unifies disparate challenges into a coherent set of solutions, represents a conceptual roadmap for practitioners. In future research, this concept could be applied to real-world scenarios through longitudinal case studies within large organisations.

## **5. Conclusion**

This study has examined the theoretical foundations and evolution of Agile methodologies, along with a comprehensive analysis of the challenges encountered in managing large development teams under high-load conditions. The research has demonstrated that traditional Agile approaches, originally designed for small teams, require significant adaptation when applied to large-scale, distributed structures. Identified issues related to insufficient coordination, diffusion of responsibility, communication barriers, and technical complexities can be effectively addressed through the implementation of integrated strategies.

The recommendations outlined in this study include the establishment of a unified high-level backlog, the development of a corporate culture centered on collaboration, continuous employee upskilling, optimization of iteration duration, active adoption of modern digital tools, and prioritized testing. The comprehensive application of these measures enhances operational efficiency, reduces time-to-market, and improves the quality of the final product.

Thus, integrating Agile approaches with traditional management methods and leveraging modern technological solutions are key factors for successfully scaling Agile methodologies under high-load conditions.

## **References**

- [1]. Scaling Agile: Strategies for Managing Large Development Teams. [Electronic resource] Access mode: <https://www.kovair.com/blog/scaling-agile-strategies-for-managing-large-development-teams/> / (date of access: 02/14/2025).
- [2]. Daraojimba E. C. et al. Comprehensive review of agile methodologies in project management //Computer Science & IT Research Journal. – 2024. – Vol. 5 (1). – pp. 190-218.
- [3]. Berbegal-Mirabent J., Gil-Doménech D., Berbegal-Mirabent N. Teaching agile methodologies in a project management course //Proceedings of the 3rd International Conference on Higher Education



- Advances. – Editorial Universitat Politècnica de València. - 2017. – pp. 312-320.
- [4]. Alves E. J., Gonçalves C. A. Agile project portfolio management as a strategic enabler in the Brazilian marketplace: five case studies and one substantive theory //International Journal of Managing Projects in Business. – 2023. – Vol. 16 (3). – pp. 475-495.
- [5]. Andreev V. V., Malozemov S. N. Development of an algorithm for selecting the optimal set of tools and techniques for Agile project management in industry and engineering. – 2020. – pp. 1-5.
- [6]. Iqbal J. et al. The effects of agile methodologies on software project management in pakistani software companies //Turkish Journal of Computer and Mathematics Education (TURCOMAT). – 2021. – Vol. 12 (3). – pp. 1717-1727.
- [7]. Pal K., Karakostas B. Software testing under agile, scrum, and devops //Agile Scrum Implementation and Its Long-Term Impact on Organizations. – IGI Global. - 2021. – pp. 114-131.
- [8]. Halani K. R., Jhaharia K. A quantitative study of waterfall and agile methodologies with the perspective of project management //Contemporary Challenges for Agile Project Management. – IGI Global. - 2022. – pp. 111-133.
- [9]. Zasornova I., Lysenko, S., Zasornov, O. Choosing scrum or kanban methodology for project management in it companies //Computer systems and information technologies. – 2022. – Vol. 4. – pp. 6-12.