

Article

Not peer-reviewed version

Managing Borderless Project Teams: PM Playbooks for Iterative Delivery

[Abhi Gaikwad](#)*

Posted Date: 6 January 2026

doi: 10.20944/preprints202601.0312.v1

Keywords: project management; distributed teams; agile methodology; global software development; iterative delivery; collaboration tools



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Managing Borderless Project Teams: PM Playbooks for Iterative Delivery

Abhi Gaikwad

Independent Researcher; ab101995i@gmail.com

Abstract

Distributed projects spanning time zones and cultures strain communication, coordination, and control, demanding project management practices that explicitly govern stakeholder alignment, information flow, and decision cadence. This paper synthesizes evidence on how iterative delivery rituals can be embedded within PM governance—linking standups, sprint reviews, and retrospectives to communication plans, risk registers, change control, and visibility dashboards—to raise predictability in global initiatives. A practical framework maps collaboration tooling (e.g., video, messaging, shared wikis) to specific PM objectives, while outlining mitigations for language barriers, cultural divergence, and trust deficits common to dispersed teams. Reported benefits include clearer requirements, faster feedback cycles, improved knowledge sharing, and higher transparency, counterbalanced by recurring risks such as time-zone friction, uneven tool access, and coordination overheads, with checklists provided for PMOs to operationalize at scale. The contribution equips project planners and delivery leads with actionable playbooks to achieve scope, schedule, and quality targets under high uncertainty—without relying on co-location.

Keywords: project management; distributed teams; agile methodology; global software development; iterative delivery; collaboration tools

1. Introduction

The globalization of software development has transformed how organizations structure their project teams, with borderless collaborations becoming increasingly prevalent in contemporary software engineering practice. This shift toward distributed team configurations introduces complex challenges that traditional project management approaches struggle to address effectively [1]. The geographical dispersion of team members across different time zones, coupled with cultural and linguistic diversity, creates significant barriers to communication, coordination, and control—three fundamental pillars of successful project execution [2].

Traditional project management methodologies, characterized by sequential development phases and extensive upfront planning, prove inadequate in distributed contexts where requirements evolve rapidly and uncertainty prevails. The rigidity of these approaches conflicts with the dynamic nature of global software development, where changing market conditions and emerging stakeholder needs demand flexibility and adaptability [3]. This misalignment has prompted organizations to explore alternative management frameworks that can accommodate the unique constraints of distributed collaborations while maintaining project governance and control.

Agile methodologies offer promising solutions to these challenges through their emphasis on iterative development, continuous feedback, and adaptive planning. The core principles of agile—as articulated in the Agile Manifesto—prioritize individuals and interactions, working software, customer collaboration, and responsiveness to change [4]. These values align closely with the needs of distributed teams, where effective communication and coordination mechanisms are paramount for success. However, implementing agile practices in globally distributed settings requires careful adaptation to address the specific constraints imposed by geographical, temporal, and cultural distances [5].

This paper examines how project management playbooks can integrate iterative delivery rituals within governance structures to enhance predictability and control in borderless project teams. We synthesize evidence from empirical studies and industrial experience reports to identify effective strategies for managing distributed agile projects. Our analysis focuses particularly on how standard agile ceremonies—such as daily stand-ups, sprint planning, reviews, and retrospectives—can be embedded within formal project management frameworks to create hybrid approaches that leverage the strengths of both disciplined governance and adaptive execution.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive analysis of communication challenges in distributed agile environments and proposes mitigation strategies. Section 3 examines coordination mechanisms and their implementation across temporal and geographical boundaries. Section 4 explores trust-building strategies for distributed teams. Section 5 presents a framework for integrating agile rituals with project management governance. Section 6 discusses tooling infrastructure requirements, and Section 7 concludes with practical recommendations for project management offices.

2. Communication Challenges in Distributed Agile Environments

Effective communication represents perhaps the most significant challenge in distributed agile projects, where the informal, face-to-face interactions that characterize co-located agile teams become difficult to replicate. Research consistently identifies communication as a critical success factor in global software development, with breakdowns in information flow directly correlating with project delays, quality issues, and stakeholder dissatisfaction [6]. The communication challenges in distributed agile environments manifest across multiple dimensions, each requiring specific mitigation strategies.

Temporal distribution creates substantial communication barriers, particularly when team members work across significantly different time zones. The limited overlap in working hours reduces opportunities for synchronous communication, forcing teams to rely more heavily on asynchronous methods that can delay problem resolution and decision-making [7]. Teams distributed across regions with 8-12 hour time differences, such as between North America and Asia, may have only 1-2 hours of overlapping work time daily, creating pressure to compress essential communications into narrow windows. This temporal misalignment not only slows information exchange but also reduces the frequency and richness of informal interactions that often spark innovation and strengthen team relationships.

Cultural and linguistic diversity introduces additional communication complexities that can impede collaboration in distributed agile teams. Team members from different cultural backgrounds may have varying communication styles, with some cultures favoring direct expression while others prefer indirect approaches [8]. These differences can lead to misunderstandings, misinterpretations of urgency, and unintended offense. Linguistic barriers further complicate communication, particularly when team members must communicate in a non-native language. The cognitive load of processing technical information in a second language can reduce participation in discussions and limit the depth of technical conversations, ultimately affecting software quality and team productivity.

The absence of rich communication channels in distributed settings represents another significant challenge. Co-located agile teams benefit from spontaneous interactions, whiteboard sessions, and non-verbal cues that facilitate shared understanding and relationship building. Distributed teams, by contrast, often rely on technology-mediated communication that filters out these contextual elements [9]. Even with advanced video conferencing systems, distributed teams miss the subtle cues and informal interactions that occur naturally in shared physical spaces. This limitation can hinder the development of shared mental models and reduce team cohesion.

Communication tool limitations pose practical challenges for distributed agile teams. The effectiveness of technology-mediated communication depends heavily on tool reliability, accessibility, and usability [14]. Teams working across different organizational boundaries may face restrictions on tool usage due to security policies or infrastructure limitations. Furthermore, the proliferation of

communication tools can create fragmentation, with discussions scattered across multiple platforms and important information becoming difficult to locate. This tool fragmentation increases cognitive load and reduces communication efficiency.

To address these communication challenges, distributed agile teams must implement multifaceted strategies that combine technological solutions with process adaptations. Establishing communication protocols that specify preferred channels for different types of communication can reduce ambiguity and ensure important information reaches the appropriate stakeholders. Regular communication audits can help identify breakdowns and refine protocols based on team feedback. Additionally, investing in communication infrastructure and ensuring all team members have access to reliable, high-quality tools is essential for maintaining effective information flow in distributed contexts.

Cultural awareness training and language support represent critical investments for mitigating communication barriers arising from diversity. Teams benefit from explicit discussions about communication norms and preferences, creating shared understanding of how different team members prefer to give and receive information [13]. Language support tools, including translation services and language training, can reduce barriers for non-native speakers and create more inclusive communication environments. These investments not only improve immediate communication effectiveness but also demonstrate organizational commitment to supporting diverse team members.



Figure 1. Communication challenges in distributed agile teams and corresponding mitigation strategies. The diagram illustrates how multiple factors contribute to communication difficulties and suggests targeted approaches for addressing each challenge.

3. Coordination Mechanisms for Distributed Teams

Coordination—the process of managing dependencies between activities and team members—presents particular challenges in distributed agile environments where traditional coordination mechanisms prove insufficient. In co-located settings, coordination occurs naturally through informal interactions and shared context. Distributed teams, however, must implement explicit coordination mechanisms to ensure activities remain aligned across geographical and temporal boundaries [2]. Effective coordination in distributed agile projects requires deliberate design of processes, roles, and artifacts that facilitate dependency management despite physical separation.

The daily stand-up meeting, a cornerstone of agile coordination, requires significant adaptation in distributed contexts. While co-located teams benefit from brief, focused face-to-face meetings, distributed teams must rely on technology-mediated alternatives that preserve the meeting's essential coordination function while accommodating geographical distribution [7]. Successful distributed stand-ups often combine video conferencing with shared digital task boards to maintain visibility of work progress and dependencies. Teams working across significant time zones may implement "follow-the-sun" stand-ups where updates are passed sequentially between sites, or may rotate meeting times to distribute the burden of off-hours participation fairly across locations.

Sprint planning and review ceremonies present additional coordination challenges in distributed settings. These critical events require intensive collaboration and negotiation to establish shared understanding of requirements and acceptance criteria. Distributed teams must allocate sufficient time for these ceremonies to accommodate the additional communication overhead introduced by technology mediation [5]. Effective distributed sprint planning often involves preparatory work to ensure all participants arrive with necessary context, structured facilitation to ensure balanced participation across sites, and explicit documentation of decisions and assumptions to prevent misunderstandings as work progresses.

Architectural coordination represents a particularly complex challenge in distributed agile development. The emergent design approach common in agile methods relies heavily on continuous technical communication and refactoring, processes that become difficult when team members lack shared context [9]. Distributed teams must implement additional mechanisms for architectural coordination, such as designated architecture owners who facilitate decision-making across sites, regular technical sync meetings focused specifically on design issues, and enhanced documentation of architectural decisions and their rationales. These mechanisms help maintain technical coherence despite physical separation of team members.

Dependency management requires heightened attention in distributed agile projects, where implicit awareness of interdependencies diminishes with geographical distance. Teams must implement explicit dependency tracking mechanisms, such as visual dependency maps or cross-team coordination meetings, to identify and address potential conflicts before they impact delivery [11]. Regular cross-site coordination meetings, sometimes called "scrum of scrums" in scaled agile contexts, provide forums for addressing dependencies and resolving conflicts across team boundaries. These meetings must be carefully facilitated to ensure balanced participation and effective decision-making despite time zone challenges.

Knowledge management systems play a crucial coordination role in distributed agile teams by preserving organizational memory and making implicit knowledge explicit. Unlike co-located teams that can rely on informal knowledge sharing, distributed teams require structured approaches to capturing and disseminating critical information [10]. Effective knowledge management in distributed contexts combines documentation practices with social structures that encourage knowledge sharing. Teams may implement practices such as pair programming across sites, rotation of team members between locations, and maintained repositories of design decisions and lessons learned to facilitate coordination through shared understanding.

The coordination overhead introduced by distribution necessitates careful attention to team structure and composition. Organizations must balance the benefits of specialization against the

coordination costs of cross-site collaboration [12]. Teams may choose to organize around features rather than components to reduce cross-site dependencies, or may implement mirroring strategies where similar team structures exist at each location to facilitate coordination. These structural decisions significantly impact coordination effectiveness and should align with overall project goals and constraints.

Table 1. Coordination Mechanisms for Distributed Agile Teams

Type	Co-Located	Distributed	Factors
Stand-ups	In person	Video + board	Same time
Planning	Group session	Prep + short calls	Clear agenda
Architecture	Ongoing talk	Lead syncs	Document decisions
Dependencies	Shared context	Visual tracker	Regular review
Knowledge	Informal	Repo + channels	Curated updates

4. Trust-Building in Geographically Dispersed Teams

Trust constitutes a fundamental element of successful team functioning, yet represents one of the most challenging attributes to establish and maintain in geographically dispersed teams. Research consistently identifies trust as a critical success factor in distributed collaborations, with high-trust teams demonstrating superior performance, enhanced knowledge sharing, and more effective conflict resolution compared to low-trust counterparts [13]. The development of trust in distributed contexts requires deliberate strategies that address the unique barriers created by physical separation and cultural differences.

The absence of frequent face-to-face interaction significantly impedes trust formation in distributed teams. In co-located settings, trust develops through repeated informal interactions that allow team members to demonstrate reliability, integrity, and benevolence. These opportunities for trust-building occur naturally through shared meals, casual conversations, and non-work interactions that reveal character and establish personal connections [7]. Distributed teams lack these organic trust-building opportunities and must therefore create structured alternatives that facilitate personal connection despite physical separation. Virtual social events, non-work communication channels, and occasional face-to-face meetings represent essential investments in trust development.

The establishment of swift trust represents a particular challenge in temporary project teams that form for specific initiatives then disband. Unlike enduring organizational relationships where trust develops over extended periods, project teams must often establish effective working relationships quickly [2]. Distributed agile teams can accelerate trust formation through clear role definitions, explicit norms for interaction, and early successes that demonstrate collective capability. Project kickoff meetings that include both task-focused and relationship-building activities help establish initial trust, while consistent delivery against commitments in early iterations reinforces perceptions of reliability and competence.

Cultural differences significantly impact trust development in globally distributed teams. Trust signals and interpretations vary across cultures, with some cultures emphasizing relationship-based trust while others prioritize competence-based trust [8]. These differences can lead to misinterpretation of intentions and behaviors, inadvertently undermining trust. Teams comprising members from multiple cultural backgrounds benefit from explicit discussions about trust expectations and behaviors, creating shared understanding of how trust will be built and maintained across cultural boundaries. Cultural training that specifically addresses trust differences helps team members interpret behaviors accurately and avoid misunderstandings.

Communication transparency plays a crucial role in trust maintenance for distributed teams. When team members lack visibility into activities and decision-making processes at other locations, suspicions and misconceptions can undermine trust [9]. Distributed agile teams enhance trust through deliberate transparency practices such as open access to project artifacts, inclusive decision-making processes, and regular communication about progress and challenges. Tools that provide visibility into work status, such as shared dashboards and burndown charts, help demonstrate reliability and create accountability across sites.

The development of interpersonal connections represents another critical trust-building strategy for distributed teams. While task-focused interactions dominate in distributed contexts, trust requires knowledge of colleagues as individuals beyond their work roles [11]. Teams can facilitate these connections through virtual coffee breaks, non-work communication channels, and personal profile sharing that highlights interests and backgrounds. Some organizations implement "virtual water cooler" platforms specifically designed to foster informal interactions, while others allocate time at the beginning of meetings for personal check-ins that strengthen relationships.

Leadership behaviors significantly influence trust development in distributed teams. Leaders who demonstrate consistency, fairness, and transparency in their interactions across sites model trusted behavior and establish norms for the entire team [10]. Distributed team leaders must pay particular attention to balanced inclusion of all locations in decision-making, equitable distribution of challenging and rewarding assignments, and public recognition of contributions from all sites. These leadership practices prevent the development of "in-group/out-group" dynamics that can severely damage trust in distributed contexts.

5. Integrating Agile Rituals with Project Management Governance

The successful implementation of agile methodologies in distributed contexts requires thoughtful integration of agile rituals with formal project management governance structures. While agile methods emphasize adaptability and team self-organization, distributed settings necessitate more explicit governance to maintain alignment across geographical and organizational boundaries [3]. This integration creates a hybrid approach that preserves agile responsiveness while providing the structure needed to coordinate distributed work effectively.

Daily stand-up meetings serve as a fundamental coordination mechanism in agile approaches, but require enhancement in distributed contexts to fulfill their governance function. Beyond the traditional focus on work progress, distributed stand-ups should explicitly address cross-site dependencies, impediments requiring escalation, and alignment with overall project objectives [5]. Effective distributed stand-ups incorporate visual management through shared digital boards, structured facilitation to ensure balanced participation across locations, and explicit documentation of action items and decisions. These enhancements transform the stand-up from a simple team sync into a governance mechanism that maintains visibility and coordination across the distributed project.

Sprint planning ceremonies provide critical opportunities for aligning team activities with project objectives in distributed contexts. The collaborative nature of sprint planning supports shared understanding of requirements and priorities, while the concrete output—the sprint backlog—serves as a commitment against which progress can be measured [2]. Distributed sprint planning requires additional time to accommodate technology-mediated communication, careful preparation to ensure all participants have necessary context, and explicit documentation of assumptions and decisions. These adaptations ensure that sprint planning fulfills its governance function of creating aligned, realistic work plans despite geographical distribution.

Sprint review and retrospective ceremonies offer structured mechanisms for governance through inspection and adaptation. The sprint review provides stakeholders visibility into progress and opportunities to adjust product direction, while the retrospective enables continuous improvement of team processes [7]. In distributed contexts, these ceremonies require careful facilitation to ensure all voices are heard despite technological and geographical barriers. Effective distributed retrospectives

often employ digital collaboration tools that enable anonymous input and parallel idea generation, overcoming participation barriers that might silence less confident team members in technology-mediated settings.

The product backlog serves as a central governance artifact in agile approaches, providing a prioritized list of work items that guides team activities. In distributed contexts, the backlog requires enhanced clarity and detail to compensate for reduced opportunities for clarification [9]. Effective distributed agile teams invest additional effort in backlog refinement, creating clear acceptance criteria and detailed descriptions that minimize ambiguity. Some teams implement "three-amigo" sessions—involving development, testing, and product perspective—across sites to ensure shared understanding of backlog items before they enter sprint planning.

Metrics and reporting require careful design in distributed agile contexts to provide visibility without undermining team autonomy. Traditional project management often emphasizes detailed tracking against predefined plans, while agile approaches favor working software as the primary progress measure [1]. Distributed contexts benefit from balanced metrics that combine traditional measures like schedule adherence with agile measures like velocity and deliverable quality. These metrics should provide sufficient visibility for governance while respecting team self-organization and avoiding excessive reporting overhead.

Risk management represents another area where agile rituals and project management governance intersect productively. While agile approaches address risk through iterative delivery and frequent adjustment, distributed contexts benefit from more explicit risk identification and mitigation planning [6]. Distributed agile teams can incorporate risk discussions into sprint planning and review ceremonies, maintaining agile responsiveness while ensuring systematic attention to potential threats. Visual risk radars and regular risk review meetings help maintain awareness of emerging risks across all locations.

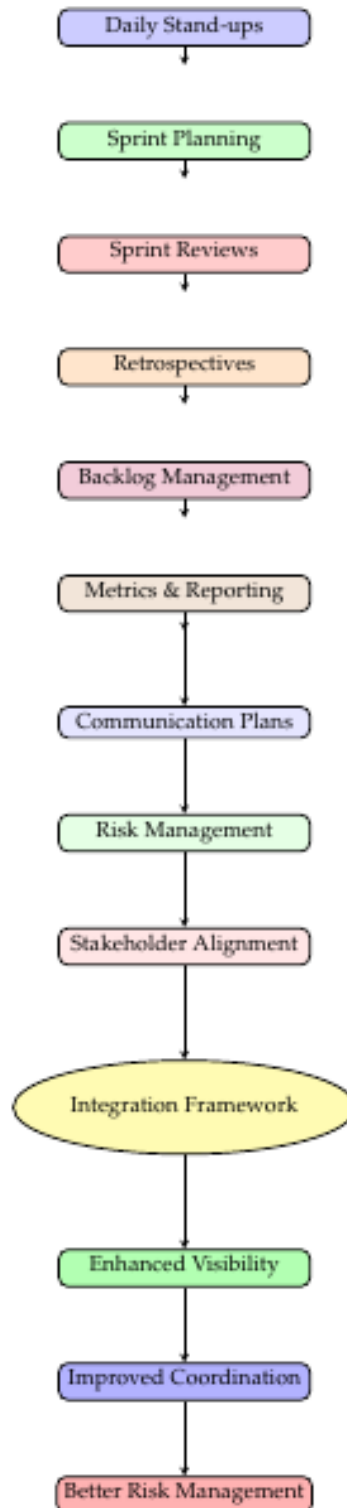


Figure 2. Framework for integrating agile rituals with project management governance in distributed contexts. The model shows how standard agile ceremonies combine with traditional PM elements to create enhanced visibility, coordination, and risk management.

6. Collaboration Tooling Infrastructure

Effective tooling infrastructure represents a critical enabler for distributed agile teams, providing the technological foundation that supports communication, coordination, and collaboration across geographical boundaries. The selection and implementation of appropriate tools significantly impact team effectiveness, with poorly chosen or integrated tools creating friction that impedes rather than

enables collaboration [14]. Distributed agile teams require a carefully curated toolset that balances functionality with usability, integration, and accessibility across different locations and contexts.

Communication tools form the backbone of distributed collaboration, enabling the frequent, rich interactions that agile methodologies require. Modern distributed teams typically employ a portfolio of communication tools addressing different interaction needs [8]. Synchronous communication tools, such as video conferencing and instant messaging, support real-time discussions and decision-making, while asynchronous tools, such as email and discussion forums, accommodate time zone differences and enable thoughtful responses. Effective teams establish clear protocols specifying which tools to use for different types of communication, reducing ambiguity and ensuring important information reaches intended recipients through appropriate channels.

Project management and tracking tools provide essential visibility into work progress and facilitate coordination across distributed team members. Agile-specific tools support key ceremonies and artifacts, including product backlogs, sprint planning, task boards, and burndown charts [5]. Effective implementation of these tools requires careful configuration to match team processes, training to ensure consistent usage, and integration with other tools in the collaboration ecosystem. The visibility provided by these tools helps compensate for the reduced situational awareness that naturally occurs in distributed settings, enabling team members to maintain understanding of overall progress and dependencies.

Version control and continuous integration systems play particularly important roles in distributed agile development by coordinating code changes and providing rapid feedback on integration issues. Distributed version control systems, such as Git, naturally support the parallel development workflows common in agile teams, while continuous integration servers automate build and test processes, alerting teams immediately when changes introduce defects [9]. These technical infrastructure elements reduce integration conflicts and provide safety nets that enable distributed teams to work independently while maintaining code quality and system stability.

Knowledge management tools help distributed teams capture and share information that would circulate informally in co-located settings. Wikis, shared documentation repositories, and decision logs preserve organizational memory and make implicit knowledge explicit [10]. Effective knowledge management in distributed contexts requires both technological solutions and social processes that encourage contribution and curation. Teams that successfully leverage knowledge management tools establish clear ownership of critical information, integrate knowledge sharing into regular workflows, and recognize contributions to shared knowledge bases.

The integration of different tools within the collaboration ecosystem significantly impacts team productivity and experience. Poorly integrated tools create friction as team members manually transfer information between systems, increasing overhead and introducing opportunities for error [7]. Well-integrated tool ecosystems, by contrast, create seamless workflows that support rather than impede collaboration. Organizations investing in distributed agile capabilities should prioritize tool integration, either through commercial integrated platforms or through custom integration of best-of-breed tools using APIs and middleware.

Tool selection and implementation require careful consideration of organizational constraints and team preferences. Security requirements, infrastructure limitations, and compliance regulations may restrict available options, while team members' prior experiences and skill levels influence adoption and effective usage [2]. Successful tool implementations involve end-users in selection decisions, provide comprehensive training and support, and allow for customization to match team workflows. These participatory approaches increase tool adoption and effectiveness, maximizing return on technology investments.

7. Conclusion and Practical Recommendations

This examination of distributed agile project management reveals both the significant challenges and substantial opportunities presented by borderless team configurations. While geographical dis-

tribution introduces complexities in communication, coordination, and trust-building, appropriate adaptations of agile practices combined with thoughtful project management governance can mitigate these challenges and leverage the benefits of diverse, globally distributed teams. The synthesis of evidence from empirical studies and industrial experience reports provides a foundation for practical recommendations that project management offices can implement to enhance distributed agile effectiveness.

Project managers leading distributed agile initiatives should prioritize communication infrastructure and protocols as foundational elements of project success. This includes investing in reliable, high-quality communication tools; establishing clear protocols for tool usage and information sharing; and creating structures that support both task-focused and relationship-building communications [8]. Regular communication audits can identify breakdowns and refinement opportunities, while communication training that addresses both technological and intercultural aspects enhances team members' capabilities to collaborate effectively across distances.

Coordination mechanisms require deliberate design in distributed contexts, with particular attention to adapting standard agile ceremonies for geographical distribution. Project managers should ensure that daily stand-ups, sprint planning, reviews, and retrospectives receive sufficient time and appropriate facilitation to overcome the limitations of technology-mediated interaction [5]. Visual management techniques, explicit dependency tracking, and regular cross-team coordination meetings help maintain alignment across sites. These coordination mechanisms should evolve based on team feedback and changing project needs.

Trust-building represents an ongoing priority rather than a one-time activity in distributed projects. Project managers should implement structured approaches to trust development, including virtual social events, non-work communication channels, and occasional face-to-face meetings when feasible [13]. Leadership behaviors that demonstrate fairness, transparency, and inclusion across all locations model and reinforce trusted interactions. Monitoring trust levels through regular check-ins and anonymous surveys helps identify and address trust issues before they significantly impact team performance.

The integration of agile rituals with project management governance creates a balanced approach that preserves agile responsiveness while providing the structure needed for distributed coordination. Project managers should focus on enhancing rather than replacing agile ceremonies with governance elements, using stand-ups for dependency management, sprint planning for alignment with project objectives, and retrospectives for continuous improvement of distributed collaboration practices [3]. This integration enables distributed agile teams to maintain both adaptability and predictability despite geographical separation.

Tooling infrastructure requires strategic attention rather than tactical implementation. Project management offices should develop comprehensive tool strategies that address communication, project tracking, version control, and knowledge management needs in an integrated manner [14]. Tool selection should involve end-users and consider organizational constraints, while implementation should include adequate training, support, and customization. Regular evaluation of tool effectiveness ensures the infrastructure continues to meet evolving team needs.

Future research should continue to explore the evolving practices of distributed agile project management, particularly as new technologies and work models emerge. Areas warranting further investigation include the impact of artificial intelligence on distributed collaboration, the evolution of hybrid work models combining remote and office-based work, and the long-term effects of distributed work on team innovation and individual well-being. This ongoing research will ensure that project management practices continue to adapt to the changing landscape of global software development.

Acknowledgments: The author acknowledges the contributions of the distributed teams whose experiences informed this research and the colleagues who provided valuable feedback on earlier versions of this framework.

References

1. H. Holmström, B. Fitzgerald, P. J. Ågerfalk, and E. O. Conchúir, "Agile practices reduce distance in global software development," *Information Systems Management*, vol. 23, no. 3, pp. 7–18, 2006.
2. I. Bose, "Lessons learned from distributed agile software projects: A case-based analysis," *Communications of the Association for Information Systems*, vol. 23, no. 1, p. 1, 2008.
3. T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1213–1221, 2012.
4. G. G. Miller, "The characteristics of agile software processes," in *Proceedings of the 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems*, 2001, pp. 385–387.
5. S. Jalali and C. Wohlin, "Agile practices in global software engineering: A systematic map," in *2010 5th IEEE International Conference on Global Software Engineering*, 2010, pp. 45–54.
6. N. K. Kamaruddin, N. H. Arshad, and A. Mohamed, "Chaos issues on communication in agile global software development," in *2012 IEEE Business Engineering and Industrial Applications Colloquium*, 2012, pp. 394–398.
7. U. Farooq and M. U. Farooq, *Exploring the Benefits and Challenges of Applying Agile Methods in Offshore Development*. Blekinge Institute of Technology, 2010.
8. S. Dorairai, J. Noble, and P. Malik, "Effective communication in distributed agile software development teams," in *Agile Processes in Software Engineering and Extreme Programming*. Springer, 2011, pp. 102–116.
9. L. Layman, L. Williams, D. Damian, and H. Bures, "Essential communication practices for extreme programming in a global software development team," *Information and Software Technology*, vol. 48, no. 9, pp. 781–794, 2006.
10. E. Hossain, M. A. Babar, and H. Y. Paik, "Using scrum in global software development: a systematic literature review," in *2009 Fourth IEEE International Conference on Global Software Engineering*, 2009, pp. 175–184.
11. C. Gurram and S. G. Bandi, "Teamwork in distributed agile software development," *International Journal of Computer Applications*, vol. 47, no. 13, pp. 1–5, 2012.
12. P. S. Taylor, D. Greer, P. Sage, G. Coleman, K. McDaid, and F. Keenan, "Do agile GSD experience reports help the practitioner?" in *Proceedings of the 2006 International Workshop on Global Software Development for the Practitioner*, 2006, pp. 87–93.
13. S. Dorairai, J. Noble, and P. Malik, "Understanding the importance of trust in distributed agile projects: A practical perspective," in *Agile Processes in Software Engineering and Extreme Programming*. Springer, 2010, pp. 172–177.
14. S. Nomura, T. Ishida, M. Yasuoka, N. Yamashita, and K. Funakoshi, "Open source software development with your mother language: Intercultural collaboration experiment 2002," in *International Conference on Human-Computer Interaction*, 2003, pp. 1163–1167.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.