ISSN : 2455-135X

AGILE VERSUS TRADITIONAL SOFTWARE DEVELOPMENT METHODOLOGIES: A CRITICAL REVIEW OF GOVERNANCE ALIGNMENT IN LARGE-SCALE ADOPTION.

NWOKPURU SAMUEL ABAFU
Department of Computing, Software Engineering,
University of Greater Manchester
San2crt@bolton.ac.uk

CHINONSO JOB
University of greater Manchester, United Kingdom.
cj5crt@bolton.ac.uk

ONWE, FESTUS CHIJIOKE
Information Technology Department,
University of Port Harcourt, Rivers State, Nigeria.
festus.onwe@uniport.edu.ng

Abstract—The comparative strengths of Agile and traditional plan-driven software development methodologies are among the most extensively documented topics in software engineering research, and a narrow restatement of that comparison offers little new to the literature. This paper instead synthesises the comparative evidence with a specific focus on a gap that recurs across the literature but is rarely addressed directly: how organisational governance structures should be adapted, rather than simply bypassed or left unreconciled, when Agile methods are adopted at scale. Drawing on quantitative project-outcome evidence (the Standish Group CHAOS Reports, Serrador and Pinto's empirical analysis), industry adoption surveys (State of Agile, PMI Pulse of the Profession), and the large-scale Agile transformation literature (Dikert et al., Rigby et al., Moe et al.), we show that the documented failure modes of large-scale Agile adoption — cultural resistance, fragmented governance, and distributed-team coordination breakdown — are predominantly governance failures rather than methodological failures of Agile itself. We synthesise the limited existing evidence on hybrid Agile–structured approaches and argue that the relevant research question is no longer "Agile or traditional" but how governance frameworks should be redesigned to support Agile delivery without forfeiting the accountability traditional methods provide, particularly in regulated and safety-critical contexts.

Index Terms—Agile software development, Waterfall, software engineering methodology, IT governance, large-scale Agile transformation, hybrid project management.

I. INTRODUCTION

The comparative evaluation of Agile and traditional, plandriven software development methodologies is one of the most heavily studied questions in software engineering practice. Sommerville's standard reference characterises traditional approaches such as the Waterfall model as well suited to projects with stable, well-understood requirements, where extensive upfront documentation and sequential phase gates support predictability and regulatory traceability [1]. The Standish Group's CHAOS Report first quantified the cost of this rigidity in the face of changing requirements, finding in 1994 that only 16% of information technology projects then surveyed succeeded fully within time, budget, and scope [2], a finding the Group's subsequent reports have continued to track and refine [3], [4].

ISSN : 2455-135X

The Agile Manifesto formalised an alternative set of values—adaptability, customer collaboration, and incremental delivery—that frameworks such as Scrum, Kanban, and

N. S. Abafu is with the Department of Computing, Software Engineering. Manuscript prepared for submission; corresponding author email to be inserted by the author prior to submission.

Extreme Programming subsequently operationalised [5], [6]. Serrador and Pinto's quantitative analysis of project outcomes provides some of the strongest empirical support for Agile's comparative advantage, finding higher success and stakeholder satisfaction rates associated with Agile practices across the projects sampled [7], and successive State of Agile and PMI industry surveys document its progressive adoption well beyond its original software-development origin [8], [9].

This comparison is, however, extensively rehearsed in the literature, and a further restatement of "Agile improves adaptability; Waterfall improves predictability" would not constitute a meaningful contribution. The more consequential and comparatively under-addressed question, which this review foregrounds, is what happens at the organisational level when Agile is adopted at scale: specifically, why large-scale Agile transformations recurringly fail not because the underlying Agile practices are unsuitable, but because the surrounding governance structures are not adapted to accommodate them [10], [11].

A. Contributions

This review contributes: (i) a synthesis of the quantitative and survey-based evidence comparing Agile and traditional methodology outcomes, situated specifically against the largescale transformation and governance literature rather than treated as a standalone comparison; (ii) an analysis distinguishing methodological limitations of Agile itself from governance-design failures that are frequently mischaracterised as methodological limitations in practice; and (iii) identification of the specific governance-design questions that remain open in the hybrid Agile–structured development literature.

B. Review Methodology

This is a structured narrative review. Sources were identified through targeted search of IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, and Google Scholar using combinations of the terms "Agile," "Waterfall," "software engineering methodology," "large-scale Agile transformation," "IT governance," and "hybrid project management," supplemented by the primary industry-evidence sources (Standish Group CHAOS Reports, State of Agile Reports, PMI Pulse of the Profession) that the comparative literature consistently treats as foundational evidence. Priority was given to peer-reviewed journal articles, systematic reviews already published on largescale Agile transformation specifically [10], and quantitative project-outcome studies [7], [12] over practitioner commentary.

The remainder of the paper proceeds as follows. Section II synthesises the comparative evidence on Agile and traditional methodology outcomes. Section III examines the large-scale Agile transformation literature through a governance lens. Section IV synthesises the limited existing evidence on hybrid approaches. Section V identifies open governance-design questions, and Section VI concludes.

II. COMPARATIVE EVIDENCE ON AGILE AND TRADITIONAL METHODOLOGY OUTCOMES

A. Quantitative Project-Outcome Evidence

The Standish Group's CHAOS Reports remain the most widely cited longitudinal evidence base for IT project outcomes, and their findings have shifted over successive editions in a way that is itself informative. The 1994 report's headline finding—only 16% full success by time/budget/scope criteria [2]—was widely interpreted as an indictment of Waterfall-era practice specifically, since Agile methods did not yet exist in formalised form. The Group's 2015 report broadened its definition of project success beyond strict time/budget/scope adherence to incorporate delivered value and stakeholder satisfaction [3], a definitional shift that itself reflects the influence of Agile values on how the industry measures success, independent of which methodology a given project used.

ISSN : 2455-135X

Serrador and Pinto's quantitative analysis directly tests the comparative hypothesis, finding that Agile-managed projects in their sample showed higher success rates and stakeholder satisfaction than comparison projects, with the effect attributed primarily to more frequent customer feedback and iterative scope refinement [7]. This is among the more methodologically rigorous quantitative comparisons in the literature, though, consistent with the limitations the authors themselves note, the sample is not representative of all industry sectors, and the comparative effect size varies considerably across the studies synthesised in later systematic reviews of empirical Agile evidence [13].

B. Industry Adoption Evidence

State of Agile survey data documents accelerating adoption through the early 2020s, with the 2020 edition specifically reporting Agile's prominence during pandemic-driven remotework disruption [14] and the 2022 edition reporting broader enterprise-strategic alignment beyond IT delivery teams specifically [15]. PMI's Pulse of the Profession similarly frames adaptive project management capability as a determinant of organisational resilience under disruption [9]. These industry surveys are valuable for documenting adoption trends but, as is well recognised in the methodology literature, carry a structural limitation: vendor- and consortium-sponsored surveys of this kind are subject to selection and success-reporting bias, since respondents with positive Agile experiences are more likely to participate and report favourably [16].

C. Where Traditional Methods Remain the Appropriate Choice

The comparative literature is consistent in identifying conditions under which traditional, plan-driven methods remain preferable rather than merely tolerated: safety-critical systems, heavily regulated domains requiring extensive audit documentation, and projects with genuinely stable, well-specified requirements [1], [17]. This is not a minor caveat; it establishes that the comparative question is conditional on project and domain characteristics rather than resolvable by a single universally superior methodology, a point the hybrid-approaches literature discussed in Section IV takes as its starting premise.

III. LARGE-SCALE AGILE ADOPTION THROUGH A GOVERNANCE LENS

A. Documented Failure Modes

Dikert et al.'s systematic literature review of large-scale Agile transformations identifies a consistent set of recurring challenges across the studies surveyed: cultural resistance to changed reporting and decision-making structures, difficulty integrating Agile teams with non-Agile organisational functions (finance, compliance, legacy IT operations), and the loss of coordination mechanisms that worked at team scale but do not naturally scale to programme or portfolio level [10]. Rigby et al.'s practitioner-oriented analysis of Agile at scale reaches a closely related conclusion from an organisational-strategy perspective: sustained executive sponsorship, not merely teamlevel Agile competence, is the determining factor in whether a large-scale transformation persists past its initial pilot phase [11]. Moe et al.'s study of Agile in distributed teams identifies a third, related failure mode: communication and trustbuilding mechanisms that Agile ceremonies assume (daily stand-ups, co-located pairing, informal hallway conversation) degrade specifically under geographic distribution, requiring deliberate compensating practices that many distributed Agile implementations do not adopt [18].

B. Reframing These Failures as Governance-Design Problems

A close reading of these three literatures supports a specific reframing: the failure modes documented are, in the majority of cases reported, failures of governance design—the absence of a redesigned decision-rights, reporting, and accountability structure compatible with Agile delivery—rather than failures of the underlying Agile practices themselves. Conboy et al.'s analysis of adoption barriers makes this point most directly, arguing that "people and culture," not technique, are the dominant determinant of adoption success, and that

organisations frequently attempt superficial process adoption (adopting Scrum ceremonies) without the accompanying governance and cultural change, producing what Misra et al. characterise as "Agile in name only"—a surface-level process change that does not alter underlying decision authority or accountability structures and consequently fails to deliver the outcomes associated with substantive Agile adoption [16], [19].

C. The Specific Compliance Tension

For regulated and safety-critical domains specifically, Turk et al.'s early but still-cited critique of agile assumptions identifies the central tension precisely: Agile's preference for working software over comprehensive documentation is in direct friction with regulatory regimes that require documentation as an audit artefact independent of whether it informed the working system [20]. This is not resolved by abandoning documentation discipline, nor by abandoning Agile's adaptive delivery; it requires a governance structure that treats documentation as a parallel compliance deliverable rather than as Agile's primary planning mechanism, a design problem the hybrid-approaches literature (Section IV) addresses directly, albeit with limited empirical validation to date.

IV. HYBRID AGILE–STRUCTURED APPROACHES: WHAT THE EVIDENCE SHOWS

A. Existing Empirical Precedent

Batra et al.'s case study of a large, distributed, strategically important software project in the cruise line industry provides one of the few empirically grounded accounts of a deliberately balanced Agile–structured approach, finding that effective practice did not sit at either methodological extreme but combined structured, milestone-based oversight at the programme level with Agile, iterative delivery at the team level [12]. Their central finding—that the appropriate balance shifts dynamically as project scale, distribution, and requirement volatility change over the project lifecycle, rather than being fixed at project initiation—is, to date, one of the most direct pieces of evidence that hybrid approaches require active governance design rather than static methodology selection.

B. Mixed Evidence on Hybrid Effectiveness

Gemino et al.'s study explicitly interrogates whether hybrid approaches represent a genuine third option or merely "a poor second choice" relative to either pure Agile or pure traditional approaches applied well, finding that hybrid approaches' success depends heavily on whether the hybridisation is a deliberate, coherent design choice or an unplanned compromise arising from incomplete Agile adoption [21]. This finding directly qualifies the optimistic framing common in industry commentary: hybridisation is not inherently beneficial, and a poorly designed hybrid can underperform a well-executed pure approach in either direction.

C. The Governance-Design Gap

Taken together, the evidence in this section establishes that hybrid Agile–structured approaches can outperform either pure approach under specific, identified conditions (large scale, high distribution, dynamic requirements), but that the literature has not yet produced a generalisable governance-design method for constructing such hybrids deliberately, as opposed to documenting successful cases after the fact. This is the specific gap this review identifies as the priority for future work, addressed further in Section V.

V. DISCUSSION: OPEN GOVERNANCE-DESIGN QUESTIONS

Synthesising Sections III and IV, we identify three open questions that the existing literature has not resolved and that future research and practice should prioritise.

1) How should decision rights be partitioned between Agile delivery teams and governance/compliance functions? Dikert et al.'s and Rigby et al.'s findings establish that unclear decision rights are a primary failure mode [10], [11], but neither study, nor the broader literature surveyed, specifies a generalisable method for partitioning decision rights that preserves both delivery agility and governance accountability.
2) How should compliance documentation be decoupled from Agile planning artefacts without becoming disconnected from the

ISSN : 2455-135X

working system it is meant to describe? Turk et al.'s documentation-friction critique [20] identifies the problem; Batra et al.'s case study [12] demonstrates one organisation's working solution, but a generalisable design pattern, rather than a single case study, remains absent from the literature.

3) Under what conditions does hybridisation outperform a well-executed pure approach, and under what conditions does it merely combine the weaknesses of both? Gemino et al.'s findings [21] demonstrate that this question has a real, non-trivial answer, but the conditions distinguishing effective from ineffective hybridisation are not yet specified with enough precision to guide exante organisational design decisions.

VI. CONCLUSION

This review has synthesised the comparative evidence on Agile and traditional software development methodologies with a deliberate focus on governance alignment, rather than restating the well-established adaptability/predictability trade-off. The evidence reviewed supports three conclusions. First, the documented failure modes of large-scale Agile transformation—cultural resistance, fragmented coordination, distributed-team breakdown—are substantially governancedesign failures rather than failures of Agile methodology itself. Second, the limited existing empirical evidence on hybrid Agile–structured approaches suggests that deliberate, dynamically adapted hybridisation can outperform either pure approach in large, distributed, requirement-volatile contexts, but that the literature has not yet produced a generalisable governance-design method for constructing such hybrids. Third, this absence constitutes the most significant and currently under-addressed gap in the comparative Agile/traditional literature, and we have identified three specific open governance-design questions that a future, more constructive contribution should address directly. A companion paper proposes one candidate governance-design framework responding to these three questions, presented together with a case-study-based evaluation protocol for the empirical testing this review shows to be currently missing from the literature.

LIMITATIONS

As a structured narrative review rather than a fully systematic review, this paper does not report PRISMA-style screening statistics. The comparative Agile/traditional literature is extremely large; this review prioritises depth on the governancealignment question identified in Section III over exhaustive coverage of the broader Agile-effectiveness literature, which is comprehensively covered elsewhere by existing systematic reviews [10], [13].

REFERENCES

[1] I. Sommerville, *Software Engineering*, 10th ed. Boston, MA: Pearson, 2015.

[2] Standish Group, "CHAOS report 1994," The Standish Group International, Inc., 1994.

[3] ——, "CHAOS report 2015," The Standish Group International, Inc., 2015.

[4] ——, "Endless modernisation: How infinite flow keeps software fresh," The Standish Group International, Inc., 2020.

[5] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, "Manifesto for agile software development," https: //agilemanifesto.org, 2001.

[6] J. Greene and A. Stellman, *Learning Agile: Understanding Scrum, XP, Lean, and Kanban*. Sebastopol, CA: O'Reilly Media, 2014.

[7] P. Serrador and J. K. Pinto, "Does agile work? — a quantitative analysis of agile project success," *International Journal of Project Management*, vol. 33, no. 5, pp. 1040–1051, 2015.

[8] E. C. Conforto, F. Salum, D. C. Amaral, S. L. da Silva, and L. F. M. de Almeida, "Can agile project management be adopted by industries other than software development?" *Project Management Journal*, vol. 47, no. 3, pp. 21–34, 2016.

[9] Project Management Institute, "Pulse of the profession 2017: Success in disruptive times," Project Management Institute, Newtown Square, PA, 2017.

[10] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, pp. 87–108, 2016.

[11] D. K. Rigby, J. Sutherland, and A. Noble, "Agile at scale," *Harvard Business Review*, vol. 96, no. 3, pp. 88–96, 2018.

[12] D. Batra, W. Xia, D. VanderMeer, and K. Dutta, "Balancing agile and structured development approaches to successfully manage large distributed software projects: A case study from the cruise line industry," *Communications of the Association for Information Systems*, vol. 27, no. 1, pp. 379–394, 2010.

[13] T. Dyba and T. Dingsøyr, "Empirical studies of agile software devel-˚ opment: A systematic review," *Information and Software Technology*, vol. 50, no. 9-10, pp. 833–859, 2008.

[14] VersionOne / Digital.ai, "State of agile report 2020," https://stateofagile. com, 2020.

[15] ——, "State of agile report 2022," https://stateofagile.com, 2022.

[16] K. Conboy, S. Coyle, X. Wang, and M. Pikkarainen, "People over process: Key challenges in agile development," *IEEE Software*, vol. 28, no. 4, pp. 48–57, 2011.

[17] K. K. Baseer, R. A. R. Mohan, and B. C. Shoba, "A systematic survey on waterfall vs. agile vs. lean process paradigms," *I-Manager's Journal on Software Engineering*, vol. 9, no. 3, pp. 34–59, 2015.

[18] N. B. Moe, D. Smite, A.˘ Sˇablis, and A.-L. B¯ orjesson, "Agile devel-¨ opment in distributed teams: Insights and practices," *Information and Software Technology*, vol. 117, p. 106129, 2020.

[19] S. C. Misra, V. Kumar, and U. Kumar, "Identifying some important success factors in adopting agile software development practices," *Journal of Systems and Software*, vol. 82, no. 11, pp. 1869–1890, 2009.

[20] D. Turk, R. France, and B. Rumpe, "Limitations of agile software processes," in *Proceedings of the Third International Conference on Extreme Programming and Flexible Processes in Software Engineering (XP2002)*, Alghero, Italy, 2002, pp. 43–46.

[21] A. Gemino, B. H. Reich, and P. M. Serrador, "Agile, traditional, and hybrid approaches to project success: Is hybrid a poor second choice?" *Project Management Journal*, vol. 52, no. 2, pp. 161–175, 2021.