INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

Comparative Study of Waterfall and Agile Methods in Software Development

Anggi Rizky Windra Putri[1*]

[1] Department of Informatics, Universitas Siber Muhammadiyah, Yogyakarta, Indonesia

ABSTRACT

This study compares Waterfall and Agile Scrum in terms of value delivery, change handling, and documentation governance through a qualitative simulation-based design. Two project archetypes are modeled: a governance intensive university library system and a volatility driven MSME mobile e-commerce application. Results show that Waterfall offers strong traceability and audit-ready control under stable conditions but delays usable value and increases the cost of late changes. Agile Scrum enables earlier value delivery and greater adaptability to evolving requirements, though it relies on stakeholder engagement and minimal documentation discipline. The findings highlight that neither method is universally superior; methodology choice should be guided by requirement volatility, governance demands, stakeholder capacity, and documentation needs.

ARTICLE INFO

Keywords:
Waterfall, Agile Scrum, SDLC, qualitative simulation, hybrid methodology

Submitted:
21/12/2025

Revision:
25/12/2025

Accepted:
01/01/2026

Published:
10/01/2026

* Corresponding Author at Department, Faculty, University, Address, City, Zipcode, Country.
E-mail address: author@email.com (author#1), author2@email.com (author#2), author3@email.com (author#3)

INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

1. Introduction

SDLC methodology choice strongly influences time-to-market, stakeholder alignment, cost predictability, and delivered quality (Mishra & Alzoubi, 2023). In practice, teams repeatedly choose between plan-driven Waterfall, which prioritizes upfront specification and formal control, and Agile Scrum, which emphasizes adaptability and early value delivery (Natarajan & Pichai, 2024). This decision requires balancing planning versus learning, documentation versus working increments, and fixed scope versus continuous reprioritization, yet practitioners often lack context-sensitive guidance that links method features to observable project outcomes (Diansyah et al., 2023; Yahya & Maidin, 2023).

The literature positions Waterfall as suitable for stable requirements and compliance-driven environments where rigorous documentation supports traceability, auditability, and maintainability (Adelakun & Iyamu, 2021). Agile methods better fit volatile settings where rapid MVP delivery and frequent stakeholder feedback enable iterative refinement (Kasauli et al., 2021). However, comparative evidence remains uneven because many studies isolate single outcomes—such as testing or cycle time—without integrating governance and organizational constraints, and the rise of hybrid approaches further indicates that binary prescriptions rarely match real project conditions (Najihi et al., 2022; Malik et al., 2022).

This study addresses these gaps by applying a qualitative, simulation-based comparative design that evaluates Waterfall and Agile Scrum across two contrasting project archetypes: a university library management system with stable requirements and formal documentation needs, and an MSME mobile e-commerce application with rapid MVP targets and requirement volatility. The simulation examines how phase structure, feedback cadence, documentation intensity, and change-handling mechanisms translate into differences in development time, value-delivery timing, flexibility, and governance artifacts (Mishra & Alzoubi, 2023; Natarajan & Pichai, 2024; Yahya & Maidin, 2023). The study contributes an integrated comparison lens that connects delivery patterns, change accommodation, and documentation practices to support holistic method selection. It also provides a transparent simulation approach to reason consistently about methodology trade-offs and derives practical implications for hybridization by identifying plan-driven elements that satisfy governance demands and Agile practices that sustain iterative delivery under change (Diansyah et al., 2023; Malik et al., 2022).

INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

2. Literature Review

2.1 Comparative Evidence on Waterfall and Agile Methodologies

Empirical work in the last five years increasingly frames Waterfall and Agile as context-fit choices rather than universally superior alternatives. Evidence from organizational transition research indicates that Agile adoption can increase delivery cadence and strengthen feedback mechanisms that enable earlier validation and more responsive planning (Natarajan & Pichai, 2024). Comparative analyses similarly suggest that Agile aligns better with volatile requirements, whereas Waterfall remains appropriate for fixed-scope initiatives under strong governance and compliance constraints where predictability and traceability are central success criteria (Mishra & Alzoubi, 2023).

Testing-focused research reinforces this distinction: plan-driven lifecycles often concentrate verification late, which delays defect discovery and amplifies rework, while Agile practices distribute testing across iterations to support earlier detection and incremental resolution that can limit technical debt growth (Najihi et al., 2022). Studies of distributed global teams further show that Waterfall's detailed documentation can improve coordination through shared reference artifacts, although this advantage weakens when teams must absorb frequent requirement change that benefits from short feedback loops and iterative reprioritization (Adelakun & Iyamu, 2021). Overall, the literature positions Waterfall as stronger in predictability and audit-ready control under stable requirements, and Agile as stronger in adaptability and incremental value delivery under continuous change and stakeholder feedback (Mishra & Alzoubi, 2023; Natarajan & Pichai, 2024).

2.2 Change Volatility and Requirements Management

Requirements volatility is a decisive contingency in SDLC methodology selection because Waterfall and Agile manage change through fundamentally different control mechanisms. Evidence from hybrid settings shows that "Agile islands" embedded in predominantly Waterfall organizations often face requirements-engineering friction, including traceability breaks and misalignment between sprint-level user stories and enterprise requirement specifications, which can translate into coordination overhead unless explicit integration practices exist (Kasauli et al., 2021). In contrast, Agile institutionalizes requirement evolution through backlog refinement, sprint reviews, and iterative reprioritization, enabling continuous incorporation of stakeholder feedback and reducing the likelihood that delivered features diverge from current needs

INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

(Natarajan & Pichai, 2024). Waterfall, however, deliberately constrains midstream change via formal baselining and change control, which can protect projects operating under strict governance, compliance, or contractual commitments where stable specifications and auditable decisions define success, even if this approach reduces responsiveness to rapidly shifting demands (Mishra & Alzoubi, 2023).

2.3 Stakeholder Involvement and Feedback Mechanisms

Stakeholder engagement differs substantially between Waterfall and Agile and directly affects alignment with user needs and satisfaction outcomes. Agile embeds continuous stakeholder input through structured roles and recurring reviews, which enables earlier validation and course correction compared with phase-gated models that typically concentrate feedback at milestone or acceptance stages (Diansyah et al., 2023; Najihi et al., 2022). Evidence from organizational transition work also indicates that frequent demonstrations and iterative refinement can translate stakeholder feedback into ongoing adjustments that would be slower and more formalized under plan-driven governance (Natarajan & Pichai, 2024).

However, Agile's engagement model depends on stakeholder availability and decision authority; distributed and hierarchical environments may struggle to sustain frequent participation, which can weaken the intended feedback advantages and shift decisions to proxies (Adelakun & Iyamu, 2021). In complex, multi-stakeholder settings, especially where coordination scales beyond a single product owner, studies report the need for hybrid governance arrangements that preserve Agile iteration while retaining structured stakeholder committees for strategic alignment (Kasauli et al., 2021).

2.4 Documentation Practices and Governance Requirements

Documentation intensity is a core differentiator between Waterfall and Agile and directly affects knowledge management, compliance readiness, and long-term maintainability. Waterfall emphasizes comprehensive artifacts—requirements and design specifications, test plans, and traceability evidence—which support auditability, contractual verification, and structured handover to maintenance teams, and these artifacts can also function as asynchronous coordination infrastructure in distributed settings (Malik et al., 2022; Adelakun & Iyamu, 2021). In contrast, Agile prioritizes working software over extensive documentation and typically relies more on executable knowledge such as code, automated tests, and lightweight artifacts, which can accelerate delivery but may reduce traceability and create maintainability risk when teams

INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

change or governance demands increase (Diansyah et al., 2023; Najihi et al., 2022). Recent work therefore emphasizes tailoring documentation to context rather than adhering rigidly to methodology dogma, showing that teams can preserve iterative delivery while adding "just-enough" documentation and compliance artifacts when project constraints require stronger traceability and long-term supportability (Natarajan & Pichai, 2024; Malik et al., 2022).

2.5 Hybrid Approaches and Context-Fit Models

Interest in hybrid SDLC approaches has grown because recent studies agree that neither Waterfall nor Agile consistently outperforms the other across all project conditions. Hybrid models aim to combine plan-driven rigor with iterative delivery by sequencing or integrating practices to match governance demands and requirement volatility, and empirical work reports that structured hybrids can improve delivery reliability when teams must balance upfront stability with ongoing adaptation (Yahya & Maidin, 2023). Research on process tailoring further supports this direction by proposing systematic frameworks that map project characteristics—such as regulatory intensity, organizational maturity, and stakeholder capacity—to appropriate process configurations, enabling hybrids to be designed deliberately rather than applied ad hoc (Malik et al., 2022). Case-based comparative evidence also indicates that hybrids can mitigate the coordination problems that emerge when Agile teams operate within Waterfall governance by establishing explicit interfaces between sprint execution and phase-gate control, thereby aligning backlogs, reviews, and enterprise requirement management (Kasauli et al., 2021; Mishra & Alzoubi, 2023).

However, the hybrid literature remains incomplete in several respects. Studies still provide limited operational guidance on which concrete practices should be combined and how teams should integrate governance artifacts with iterative workflows without degrading delivery speed (Diansyah et al., 2023). Empirical evidence on hybrid effectiveness also remains uneven because rigorous comparisons that isolate the contribution of specific hybrid configurations are uncommon, which restricts defensible claims about causal impact (Natarajan & Pichai, 2024). In addition, organizational and cultural constraints continue to shape hybrid adoption, especially in environments with entrenched plan-driven processes or limited Agile maturity, where coordination and requirements engineering challenges can persist without explicit integration design (Kasauli et al., 2021). These limitations motivate the present study to establish a clearer

INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

baseline by comparing Waterfall and Agile Scrum under controlled project archetypes before deriving implications for context-fit hybridization (Yahya & Maidin, 2023; Malik et al., 2022).

3. Methodology

3.1 Research Design and Rationale

This study applies a qualitative, simulation-based comparative design to evaluate Waterfall and Agile Scrum across two contrasting project archetypes, using structured representations of practices, decision points, and governance artifacts derived from documented methodology rules and comparative evidence. The approach supports controlled comparison of methodological mechanisms while reducing confounding influences common in retrospective case studies, such as organizational heterogeneity, differences in team capability, and external disruptions (Mishra & Alzoubi, 2023). Rather than producing stochastic estimates or statistically modeled outputs, the simulation emphasizes traceable reasoning about how each method's structure shapes development trajectories and observable outcomes, which suits exploratory aims focused on explaining mechanisms rather than estimating population-level effect sizes (Malik et al., 2022).

The design proceeds by defining two project archetypes, specifying Waterfall and Scrum implementations with explicit assumptions about practices and artifacts, simulating development trajectories to capture phase sequencing, deliverable timing, change accommodation, and documentation outputs, and then comparing outcomes thematically across three criteria: development time and value delivery patterns, flexibility and change accommodation, and documentation practices and governance artifacts. The study prioritizes internal validity through transparent assumptions and logical consistency, while it limits claims to mechanism-oriented insights due to constrained external generalizability, consistent with theory-building research that can motivate later empirical validation (Natarajan & Pichai, 2024).

3.2 Project Archetypes

Two project archetypes represent contrasting contextual demands that the literature associates with different methodology fit. The first archetype, a University Library Management System (ULMS), models an integrated enterprise system for cataloging, circulation, user administration, fines, and repository access, and it reflects conditions commonly aligned with plan-driven development: relatively stable and well-understood requirements, strong governance and compliance expectations, and a need for comprehensive documentation to support coordination

INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

and long-term maintenance, with operational value typically realized after full module integration (Adelakun & Iyamu, 2021; Diansyah et al., 2023).

The second archetype, a Mobile E-Commerce Application for MSMEs (UMKM), models a consumer-facing product that supports storefront creation, listings, transactions, and customer engagement, and it reflects conditions commonly aligned with Agile Scrum: evolving requirements driven by market dynamics and user feedback, pressure for early MVP delivery, iterative release cadence, sustained stakeholder involvement, and "just-enough" documentation that prioritizes maintainability and integration over exhaustive specifications (Kasauli et al., 2021; Natarajan & Pichai, 2024; Yahya & Maidin, 2023).

3.3 Methodology Specifications and Assumptions

Waterfall was specified as a sequential, phase-gated process with explicit governance and documentation controls: requirements analysis for 12 weeks using stakeholder workshops, document review, and use-case modeling to produce a signed-off requirements specification and traceability artifacts; system design for 10 weeks covering architecture, database and interface design, and detailed modeling with formal review; implementation for 24 weeks based on approved designs with standards-based code review and unit testing targets; integration and testing for 10 weeks including functional verification against requirements, performance and security checks, and user acceptance testing with formal defect tracking; and deployment and transition for 6 weeks covering production rollout, training, documentation handover, and initial support, yielding a total simulated duration of 62 weeks with primary value realization concentrated at deployment, while midstream changes follow formal change control with assumed review and impact-analysis overhead (Adelakun & Iyamu, 2021; Malik et al., 2022).

Agile was specified as a Scrum-based iterative process with two-week sprints: an initial Sprint 0 for product visioning, architectural spiking, and backlog formation, followed by 20 development sprints with planning, daily coordination, continuous development and testing, stakeholder reviews, and retrospectives; the simulation assumes MVP delivery by Sprint 4 with subsequent incremental releases on a fixed cadence through backlog reprioritization at sprint boundaries, followed by a final hardening sprint for integration testing and production readiness, resulting in a total simulated duration of 42 weeks with value delivery beginning at MVP and continuing through iterative enhancements; change accommodation occurs primarily through backlog reprioritization rather than formal change control, while within-sprint changes

INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

remain constrained by team agreement to preserve sprint commitment (Natarajan & Pichai, 2024; Yahya & Maidin, 2023)..

3.4 Evaluation Criteria

The comparison used three evaluation criteria. Development time and value-delivery pattern assessed total duration, time-to-first-value, and whether value delivery concentrates at the end or emerges incrementally; in the simulation, the Waterfall ULMS scenario delivers first operational value at project completion, whereas the Agile UMKM scenario delivers an MVP earlier and then provides additional increments on a fixed cadence, reflecting the trade-off between upfront planning and early learning through partial delivery (Natarajan & Pichai, 2024).

Flexibility and change-accommodation cost examined the procedural and temporal overhead of incorporating requirement changes across stages; Waterfall routes changes through formal impact analysis and approval mechanisms that raise late-change costs, while Agile primarily absorbs change through backlog reprioritization at sprint boundaries while discouraging within-sprint scope shifts to protect sprint commitment, consistent with evidence that volatility and governance strongly mediate method fit (Kasauli et al., 2021; Mishra & Alzoubi, 2023). Documentation practices and governance artifacts evaluated the volume, formality, and intent of produced documentation; Waterfall emphasizes comprehensive specifications and traceability to support auditability and knowledge transfer, whereas Agile emphasizes lean, value-focused artifacts supplemented by iterative records and executable specifications, highlighting a trade-off between documentation overhead and governance-oriented maintainability (Adelakun & Iyamu, 2021; Diansyah et al., 2023).

3.5 Simulation Procedure and Analysis

The simulation applied methodology rules to each project archetype through structured reasoning across four pairings—ULMS–Waterfall, ULMS–Agile, UMKM–Waterfall, and UMKM–Agile—by defining initial conditions, stepping through phases or sprints, introducing contingencies such as requirement change, stakeholder feedback, and technical risk, and documenting the resulting development trajectory in narrative form, including assumed durations, value-delivery timing, change-handling mechanisms, and produced artifacts; the procedure iterated three times to refine assumptions and ensure logical consistency with the methodology definitions and documented practices in the literature (Malik et al., 2022; Natarajan & Pichai, 2024; Yahya & Maidin, 2023).

INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

The study then analyzed simulation narratives using thematic qualitative analysis by coding segments against the evaluation criteria—development time and value delivery, flexibility and change accommodation, and documentation and governance artifacts—and comparing patterns across methods and archetypes to identify consistent mechanisms and context-specific implications (Mishra & Alzoubi, 2023). To strengthen credibility, the interpretation triangulated simulated patterns with reported empirical observations in the literature, using prior findings on release cadence, requirements engineering friction, and governance constraints as plausibility checks for the simulation outcomes (Kasauli et al., 2021; Natarajan & Pichai, 2024).

3.6 Rigor and Limitations

Rigor was strengthened through four measures. First, the study ensured transparency by explicitly documenting methodology specifications, archetype characteristics, and the reasoning steps used to derive each simulated trajectory, enabling readers to evaluate internal logic and reproduce the simulation under alternative assumptions (Malik et al., 2022). Second, it maintained traceability by linking simulated outcomes to stated methodology rules and the supporting literature, reducing reliance on unverifiable researcher intuition.

Third, it applied triangulation by checking whether key simulated patterns aligned with empirical observations reported across multiple studies on distributed work, requirements engineering frictions, and Agile transition outcomes (Adelakun & Iyamu, 2021; Kasauli et al., 2021; Natarajan & Pichai, 2024). Fourth, it controlled interpretation by explicitly limiting claims to mechanism-oriented, exploratory insights, acknowledging that the simulation abstracts away factors that materially shape real projects—such as team capability variation, organizational culture, and technical complexity—thereby constraining external validity to contexts resembling the defined archetypes and positioning results as hypothesis-generating rather than definitive evidence (Mishra & Alzoubi, 2023).

4. Result

This chapter reports the outcomes of a structured simulation designed to clarify how Waterfall and Agile (Scrum) perform under contrasting project conditions. The simulation compares two representative scenarios: an integrated university library management system modeled with Waterfall and a mobile e-commerce application for MSMEs modeled with Agile Scrum. The analysis evaluates three criteria that directly influence project success in practice: development

INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

time and value-delivery pattern, flexibility toward requirement change, and documentation practices.

4.1. Simulated Project Scenarios

4.1.1. Waterfall Project: Integrated Library Management System

The simulation models a library management system for a large university. The system includes catalog management, lending and returns, member administration, inventory tracking, and statistical reporting. The scenario assumes stable and well-defined requirements, driven by administrative governance and archival compliance. These characteristics represent environments where the organization prioritizes formal approval processes, extensive documentation, and controlled change handling.

4.1.2. Agile Project: Mobile E-commerce Application for MSMEs

The simulation models a mobile e-commerce product for MSMEs that sell handicrafts. The initial scope targets an MVP that supports product browsing, shopping cart, checkout, and order notification. The scenario assumes evolving requirements due to market trends and continuous user feedback. The model therefore prioritizes rapid time-to-market, incremental feature delivery, and iterative validation of user value.

4.2. Method Implementation in the Simulation

4.2.1. Waterfall Implementation

The simulation applies Waterfall through sequential phases with gate-based progression and strict requirement baselining.

- **Requirements Analysis (3 months):** the team elicits requirements from stakeholders and produces an SRS that serves as the authoritative scope baseline.
- **Design (4 months):** the team defines system architecture, database design, interface specifications, and module designs, then validates them through formal reviews.
  Implementation (6 months): the team develops modules according to approved designs and integrates components near the end of the phase.
- **Testing (3 months):** the team executes unit, integration, system, and UAT activities, then resolves defects before release approval.

INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

- **Deployment (1 month):** the team deploys the system to production and completes operational configuration.
- **Maintenance (continuous):** the team provides defect fixes and minor enhancements after go-live.

This configuration represents a plan-driven lifecycle that concentrates major user-facing delivery toward the end of the development cycle.

4.2.2. Agile Scrum Implementation

The simulation applies Agile using Scrum with iterative planning, development, review, and retrospective cycles.

- **Concept and Inception (1 month):** the team defines product vision, identifies initial user stories, and prioritizes a product backlog for MVP delivery.
- **Sprints (10 sprints, 2 weeks each):**
    - the team selects prioritized user stories during sprint planning,
    - develops and tests features continuously during the sprint,
    - demonstrates increments in sprint review to gather stakeholder feedback, and
    - improves team practices through sprint retrospective.
- **Release cadence (every 2–4 sprints):** the team releases functional builds to users or beta testers for early validation.
- **Maintenance (continuous):** the team fixes defects and extends features based on iterative feedback.

This configuration represents an adaptive lifecycle that delivers usable product increments early and repeatedly.

4.3. Simulation Results

4.3.1. Development Time and Value Delivery Pattern

The simulation differentiates the two methodologies most clearly through the timing of user value delivery. Waterfall produces a linear schedule that accumulates deliverables across phases and typically delivers a fully functional product near the end of the lifecycle. Agile delivers a

INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

working increment early and expands functionality through short iterations, thereby accelerating time-to-market for an MVP.

Table 4.1. Simulated development time and value delivery pattern

Methodology	Project Scenario	Estimated Duration	Value Delivery Pattern
Waterfall	Integrated Library System	~17 months (3+4+6+3+1)	Full operational value concentrates near end-of-cycle release
Agile (Scrum)	Mobile E-commerce (MVP)	~5 months (1 month inception + 10×2-week sprints)	Usable increments emerge early; features expand every sprint

Table 4.1 shows that Agile achieves a shorter timeline for MVP availability because it distributes development and validation across repeated sprints. Table 4.1 also shows that Waterfall provides a clearer phase-based plan but delays operational value until completion of implementation and full testing. The simulation therefore attributes time-to-market advantages to Agile's iterative release strategy rather than to a reduction in total engineering effort.

4.3.2. Flexibility Toward Requirement Change

The simulation evaluates flexibility as the ability to incorporate requirement changes without triggering disproportionate rework. Waterfall controls change through requirement baselining and formal approvals. This approach supports predictability when requirements remain stable but increases the cost of change once downstream phases begin. Agile treats change as a managed input by reprioritizing the backlog and scheduling modifications into subsequent sprints.

Table 4.2. Simulated flexibility and change implications

Aspect	Waterfall	Agile (Scrum)
Handling requirement change	Restricts change after SRS baseline; relies on change control	Accepts change via backlog reprioritization

INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

Aspect	Waterfall	Agile (Scrum)
Rework impact	High when change appears late; may cascade across design, implementation, and test	More contained because change enters future sprint scope
Stakeholder involvement	Intensive at initiation and UAT gates	Continuous through sprint reviews and feedback loops

Table 4.2 shows that Waterfall reduces uncertainty by freezing scope early, which suits regulated or compliance-driven contexts. Table 4.2 also shows that Agile sustains responsiveness by integrating feedback into sprint planning, which suits environments where market and user needs evolve during development. The simulation therefore positions flexibility as a function of governance structure and feedback cadence rather than as an intrinsic superiority of one method.

4.3.3. Documentation Practices and Traceability

The simulation compares documentation through its completeness, timing, and operational purpose. Waterfall emphasizes comprehensive upfront documentation that supports traceability, auditability, and structured handover. Agile produces leaner documentation that evolves alongside the product and prioritizes artifacts that directly support implementation and coordination.

Table 4.3. Simulated documentation characteristics

Aspect	Waterfall	Agile (Scrum)
Documentation style	Comprehensive, formal, upfront	Selective, "just enough," evolves iteratively
Primary artifacts	SRS, detailed design, test plans, UAT documentation	Product backlog, user stories, sprint artifacts, targeted technical notes
Operational effect	Strengthens audit, compliance, and maintainability	Accelerates iteration but depends on team discipline for consistency

Table 4.3 shows that Waterfall strengthens traceability and supports governance demands through structured documentation. Table 4.3 also shows that Agile improves delivery speed by

INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

limiting documentation to high-value artifacts, but it requires sustained discipline to prevent knowledge loss and to maintain maintainability over time.

4.4. Consolidated Interpretation of Findings

The simulation indicates that method suitability depends primarily on project context. Waterfall performs effectively when requirements remain stable, formal approvals matter, and the organization values strong traceability and controlled change. Agile Scrum performs effectively when requirements evolve, early market entry matters, and stakeholders can engage continuously to refine priorities. The results therefore support a context-fit interpretation: neither methodology dominates across all criteria, and project characteristics determine which trade-offs produce the most reliable outcomes, as evidenced in Tables 4.1–4.3.

5. Conclusion and Suggestion

This study compared Waterfall and Agile Scrum through a qualitative, simulation-based design using two contrasting project archetypes—a governance-intensive university library management system and a volatility-driven MSME mobile e-commerce application—and found that neither methodology is universally superior. The simulated outcomes indicate that Waterfall best supports contexts requiring stable requirements, strong auditability, and comprehensive documentation, but it concentrates value delivery near project completion and incurs high coordination overhead when late changes occur. In contrast, Agile Scrum delivers usable value earlier through incremental releases, absorbs evolving requirements more efficiently via backlog reprioritization, and strengthens feedback-driven refinement, although it depends on sustained stakeholder availability and disciplined lightweight documentation to preserve traceability and long-term maintainability. Overall, the findings reinforce a context-fit perspective: practitioners should select or hybridize methods based on requirement volatility, governance and compliance demands, stakeholder engagement capacity, and documentation needs, and future research should validate these mechanism-level insights with empirical studies that test specific hybrid configurations across diverse organizational settings.

INTERNATIONAL JOURNAL OF BUSINESS ECONOMICS AND INFORMATICS
Volume 01, Issue 01, December 2025, pp. 57 – 72
https://ejournal.aropress.org/index.php/IJOBEI/index
DOI:

Reference

Adelakun, O., & Iyamu, T. (2021). Translation of activities in a global virtual teams software development: Agile vs. Waterfall. *Journal of Cases on Information Technology, 23*(4), 1–19. https://doi.org/10.4018/JCIT.20211001.OA11

Diansyah, A. F., Abdul Rahman, M. N., Handayani, R., Nur Cahyo, D. D., & Utami, E. (2023). Comparative analysis of software development lifecycle methods in software development: A systematic literature review. *International Journal of Advances in Data and Information Systems, 4*(2), 95–110. https://doi.org/10.25008/ijadis.v4i2.1295

El-Sokkary, N., El-Masry, W. H., & Darwish, N. R. (2021). A proposed hybrid approach for developing healthcare information systems. *International Journal of Computer Applications, 183*(27), 37–44. https://doi.org/10.5120/IJCA2021921664

Făgărășan, C., Popa, O., Pîslă, A., & Cristea, C. (2021). Agile, waterfall and iterative approach in information technology projects. *Annals of the University of Oradea, Economic Science Series, 30*(2), 76–85. https://doi.org/10.15660/auofmte.2021-2.3592

Faruk, M. J. H., Subramanian, S., Shahriar, H., Valero, M., Li, X., & Tasnim, M. (2022). Software engineering process and methodology in blockchain-oriented software development: A systematic study. In *Proceedings of the IEEE/ACIS International Conference on Software Engineering Research, Management and Applications (SERA)* (pp. 1–8). IEEE. https://doi.org/10.1109/SERA54885.2022.9806817

Fujita, T. (2023). Reconsideration and proposal of development models in projects: "Quasi" development models: quasi-waterfall and quasi-agile. *European Journal of Social Sciences Studies, 9*(2), 45–62. https://doi.org/10.46827/ejsss.v9i2.1575

John, J. J., & Sharma, S. S. (2024). A comparative study of agile and waterfall software development methodologies. *International Journal of Advanced Research in Science, Communication and Technology, 4*(1), 458–463. https://doi.org/10.48175/ijarsct-15207

Kasauli, R., Knauss, E., Nakatumba-Nabende, J., & Kanagwa, B. (2021). Agile islands in a waterfall environment: Requirements engineering challenges and strategies in automotive. In *Proceedings of the 14th Innovations in Software Engineering Conference (ISEC)* (pp. 1–11). Association for Computing Machinery. https://doi.org/10.1145/3383219.3383223

Malik, T., Zhao, Y., Gopsill, J. A., McAlpine, H. C., Chan, W. M., & Hicks, B. J. (2022). A framework for the construction and tailoring of engineering development process models. *IEEE Transactions on Engineering Management, 71*, 1459–1473. https://doi.org/10.1109/TEM.2021.3132699

Menon, V., Sinha, R., & MacDonell, S. (2021). Architectural challenges in migrating plan-driven projects to agile. In *Proceedings of the 10th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)* (pp. 223–228). SCITEPRESS. https://doi.org/10.5220/0005383502230228

Mishra, A., & Alzoubi, Y. I. (2023). Structured software development versus agile software development: A comparative analysis. *International Journal of Systems Assurance Engineering and Management, 15*(5), 1801–1815. https://doi.org/10.1007/s13198-023-01958-5

Mokhtar, R., & Khayyat, M. (2022). A comparative case study of waterfall and agile management. *SAR Journal: Science and Research, 5*(1), 44–50. https://doi.org/10.18421/sar51-07

Najihi, S., Elhadi, S., Ait Abdelouahid, R., & Marzak, A. (2022). Software testing from an agile and traditional view. *Procedia Computer Science, 203*, 775–780. https://doi.org/10.1016/j.procs.2022.07.116

Natarajan, T., & Pichai, S. (2024). Transition from waterfall to agile methodology: An action research study. *IEEE Access, 12*, 50267–50284. https://doi.org/10.1109/ACCESS.2024.3384097

Pargaonkar, S. (2023). A comprehensive research analysis of software development life cycle (SDLC) agile & waterfall model advantages, disadvantages, and application suitability in software quality engineering *[Preprint]*. OSF Preprints. https://doi.org/10.31219/osf.io/m7yq4

Rahman, A. (2024). Agile project management: Analyzing the effectiveness of agile methodologies in IT projects compared to traditional approaches. *American Journal of Business and Information Systems, 4*(4), 98–108. https://doi.org/10.69593/ajbais.v4i04.127

Saeedi, K., & Visvizi, A. (2021). Software development methodologies, HEIs, and the digital economy. *Education Sciences, 11*(2), Article 73. https://doi.org/10.3390/EDUCSCI11020073

Umer, M. M. (2025). A practical implementation of customized Scrum-based agile framework in aerospace software development under DO-178C constraints *[Preprint]*. *arXiv*. https://arxiv.org/abs/2511.14215

Vatan, E., Raissi Ardali, G. A., & Shahin, A. (2022). Selecting information systems development models based on organizational culture: An integrated approach of DEMATEL and ANP. *VINE Journal of Information and Knowledge Management Systems, 54*(3), 672–698. https://doi.org/10.1108/vjikms-08-2021-0164

Yahya, N., & Maidin, S. S. (2023). Hybrid agile development phases: The practice in software projects as performed by software engineering team. *Indonesian Journal of Electrical Engineering and Computer Science, 29*(3), 1738–1749. https://doi.org/10.11591/ijeecs.v29.i3.pp1738-1749