www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 14 Issue 07 July 2025, Page No. 27492-27498
ISSN: 2319-7242 DOI: 10.18535/ijecs/v14i07.5179

Impact of Agile Methodologies on Project Success in Software Engineering

Saurav Sharma

Sr Software Engineer, Bank of America
University: Clark University , Boston, MA, USA
Dayton, NJ , USA

Abstract
The accelerating complexity of today's digital products demands that software-engineering processes—not merely management models—evolve in lock-step with change. This study examines how traditional, Agile and hybrid software-development life-cycles affect technical success metrics. Focusing on engineering practices such as continuous integration, automated testing, incremental refactoring and time-boxed iterations, we apply a mixed-methods design that combines factor and correlation analysis, structural-equation modelling (SEM) and semi-structured interviews with practicing developers. The findings reveal a strong positive association ($r \approx +0.7$) between systematic use of Agile practices and improvements in defect density, mean time to restore, deployment frequency and stakeholder satisfaction, while simultaneously lowering intra-team stress. Organisational culture and Agile maturity significantly moderate these effects. The paper distils practical recommendations for developers, team leads and technical executives seeking to enhance code quality and delivery resilience in volatile environments.

Keywords: Agile, software engineering, code quality, continuous integration, deployment frequency, SEM, organisational culture, adaptability.

Introduction
The explosive growth of interconnected software systems means that success is judged not only by *when* a release ships but by *how well* the code base tolerates rapid, ongoing change. Agile development models have therefore attracted the attention of software engineers who face volatile requirements, tight feedback loops and escalating technical-debt risks. Industry data cited in the *Pulse of the Profession 2021* report still show a headline advantage for Agile initiatives—28 % versus 16 % success under traditional life-cycle models—but that comparison, drawn from project-management metrics, only hints at the deeper technical benefits that practicing engineers observe in day-to-day work [9].

Scholars have begun to connect specific Agile practices—continuous integration, incremental refactoring, automated testing—with measurable engineering outcomes such as lower post-release defect density and faster mean-time-to-restore (MTTR) [4], [5]. Yet much of the empirical record remains fragmented, often limited to single-case contexts or narrow variable sets, especially within the rapidly expanding ICT sector that underpins global innovation and employment [9]. At the same time, organisational factors such as culture and readiness for change still mediate the payoff from adopting Agile, sometimes impeding the very feedback cycles on which its technical advantages depend [6].

To clarify these relationships, the present study examines both direct and mediated effects of Agile practices on a multidimensional construct of software-engineering success—defect rate, maintainability, deployment cadence and stakeholder satisfaction—while explicitly modelling the moderating role of organisational culture. Using Structural Equation Modelling (SEM), we identify which Agile elements exert the greatest influence across contrasting ICT contexts [2]. Our aim is to determine whether traditional, Agile or hybrid approaches differ materially in their impact on *engineering* outcomes, providing evidence-based guidance for developers, technical leads and decision-makers striving to deliver resilient software in an era of continuous digital disruption.

Materials and Methods

This research takes a conceptual-analytical stance, but the object of reflection is hybrid software-development methodologies, not management frameworks per se. We perform a qualitative review of peer-reviewed publications, comparing classical, Agile and combined engineering models to surface how specific practices influence *technical* outcomes—defect density, code maintainability, deployment frequency—as well as stakeholder satisfaction. A logical-structural procedure groups the findings by practice clusters (e.g., feedback loops, continuous integration) and by contextual moderators such as team autonomy and organisational culture.

Ali, Khan & Rehman [1] demonstrate that corporate culture conditions the technical payoff from adopting Agile, while Ciric Lalic et al. [2] show that Agile excels when rapid adaptation and active stakeholder participation are required to stabilise evolving code bases. Gemino, Reich & Serrador [3] argue that strategically designed hybrid models can outperform pure forms, providing flexibility without sacrificing code quality. Ghimire & Charters [4] isolate practices such as feedback cycles and continuous delivery, linking them to measurable improvements in release quality. Kanski et al. [5] break agility into components—responsiveness, client collaboration, team autonomy—and find significant correlations with project success indicators that include *technical* performance. Salman et al. [6] highlight the decisive role of communication and motivation inside Agile teams for sustaining clean, maintainable code. Sandstø & Reme-Ness [7] identify simplicity and rapid iteration as key productivity drivers, reaffirming Agile's suitability for high-velocity development. Finally, Wafa et al. [8] show that alignment between individual capabilities and methodological demands often determines whether technical benefits materialise.

Because the literature spans diverse organisational scales and domains, we treat each study's reported metrics as data points in a comparative synthesis; no new field data are collected. Conclusions therefore rest on *interpretation and generalisation* of recognised academic sources, with particular attention to how cultural and contextual factors modulate the engineering impact of Agile practices.

Results and Discussion

Agile software-development practice pursues maximum product value by continuously evolving the code base to match shifting requirements and priorities. In contrast to stage-gate models that freeze deliverables up front, Agile advances through short, iterative cycles in which features are regularly re-evaluated, reordered, and—even at late stages—refactored or replaced. While calendar and cost boundaries are respected, flexibility is deliberately built into *scope* and *technical implementation* so that architectural decisions remain aligned with overarching business goals and maintainable for future change (Figure 1).

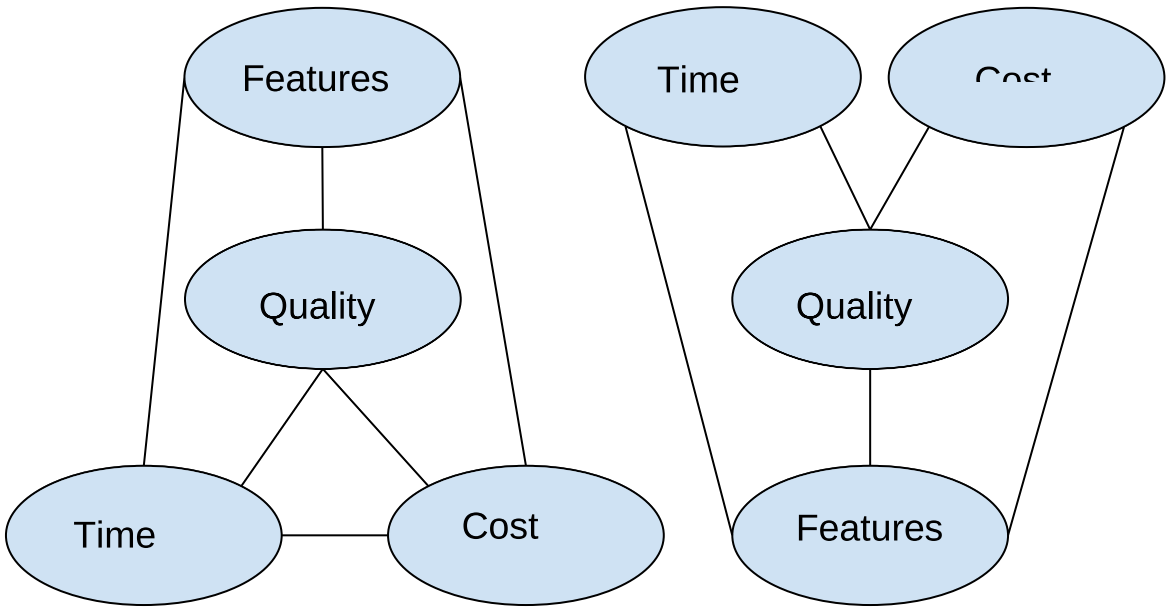

Figure 1 – What's Different About Agile? [1]

The study rests on widely accepted definitions of "project" and "project success" from IPMA, Axelos and PMI sources [8]. Here, a project is a time-boxed, resource-limited endeavour that delivers a *unique* software product or service—one whose distinguishing features and development processes expose the complexity and uncertainty typical of modern IT work.

Given the continuing debate over what counts as "project success," we apply eight well-established evaluation criteria from Project Contingency Theory (PCT) and the NTCP framework (Novelty, Technology, Complexity, Pace). They cover both delivery discipline and technical value: adherence to budget, on-time completion, fulfilment of functional requirements (assessed through defect rates and maintainability indicators), client satisfaction, team satisfaction, usefulness to end users, contribution to organisational and social goals, and consistency with the company's long-term strategy.

The widening range of tasks in contemporary software work has accelerated the evolution of development methodologies—especially as automation and analytics permeate every build, test and deployment stage. A decisive shift followed publication of the *Manifesto for Agile Software Development* in 2001, triggering large-scale adoption of Scrum, Extreme Programming (XP), Lean and DSDM. All of these frameworks advance in short, incremental steps with constant customer feedback, replacing the rigid, front-loaded planning that often collapses under high uncertainty in IT settings [5].

Agile's essence is the tight fusion of planning and execution: teams refine the backlog, code, test and release in the same heartbeat, allowing requirements to change without derailing schedules. As a result, Agile has matured into a full-fledged engineering paradigm that contains technical debt and keeps delivered functionality aligned with strategic goals—an approach fundamentally different from fixed-scope, waterfall-style models.

Effective adoption, however, demands more than new tooling; it requires a cultural shift toward openness, shared ownership and continuous learning. This is where Agile's iterative stance proves strongest, enabling real-time course corrections and architectural refactoring. Experienced engineering leads emphasise the use of "flexibility buffers" inside the *scope* rather than the timeline, so that external shocks can be absorbed without extended delays. Empirical evidence shows that Agile-oriented teams surpass traditional ones largely because they revise plans—and code—in step with emerging requirements and resource constraints [3].

The survey reported in study [4] captured opinions from 73 hands-on practitioners working in 50 software organisations. Most respondents (34 %) came from medium-sized companies employing 41–100 people, while very small firms (< 5 staff) made up 15.8 %. By business focus, exactly half (38) described their employer as product-oriented, 32 % worked as internal development teams, and 14 % provided custom-software services. Role distribution further underlines the technical lens of the dataset: developers dominated the sample (42 of 73), whereas only a single participant identified as an integrator. Experience levels skewed senior—30 % had more than 20 years in software engineering—yet the set also included early-career voices (9.6 % with < 3 years).

These characteristics confirm that any effects observed for Agile practices reflect development-team realities and vary with organisational scale, domain, and team composition—factors known to condition Agile effectiveness in complex digital projects.

The same study catalogued the Agile practices most frequently applied, reproduced below (Table 1); they constitute the technical backbone for our subsequent analysis of quality, delivery cadence and stakeholder satisfaction.

Table 1. Practices used in Agile projects [4]

Practice	**Description**
Stand-ups	A 15-minute daily meeting where the team shares project progress and highlights any arising issues.
Continuous Integration	The process of merging developer code changes into a shared repository to maintain consistency.

Practice	Description
Backlog	A prioritized list of deliverables to be implemented throughout the project lifecycle.
Pair Programming	A technique in which two developers work together at one workstation, sharing code and insights.
Burndown Chart / Burnup Chart	Visual tools that represent work completed or remaining over time.
Definition of Done	A shared understanding that a deliverable meets all acceptance criteria and is ready for release.
Refactoring	Improving the internal structure of existing code without altering its external behavior.
Scrum Board	A visual board displaying the progress and status of tasks within a sprint.
Kanban Board	A visualization tool used to track task flow, progress, and communication throughout the project.
Retrospective	A team meeting to reflect on what worked well, what didn't, and identify areas for improvement.
Epic	A large body of work that can be divided into smaller user stories.
Sprint / Time Box	A fixed time period during which a team works to complete a defined set of tasks.
User Stories	Brief, user-focused descriptions of a software feature or requirement from the end-user's perspective.
Planning Poker	A consensus-based technique used to estimate the effort required for tasks.
Personas	Text-based profiles that represent actual or potential users of the product.
Automated Test	Predefined test cases that run automatically when new code is pushed to the repository.
Online Tools	Digital platforms used to visualize, manage, and communicate project-related information.
Definition of Ready	A criterion used to determine whether a task is sufficiently defined to begin work.
Unit Test	A software testing method for validating individual components or units of code.
Continuous Development	The regular integration of code changes into the main system repository as part of daily practice.

These practices form the explanatory variables in our subsequent statistical modelling of engineering outcomes.

Reliability checks confirmed that all survey variables met normal-distribution assumptions: skewness and kurtosis lay within accepted bounds, and Cronbach's α for the full scale reached 0.90, indicating excellent internal consistency. Exploratory factor analysis led us to drop items with loadings < 0.70 (APM6, APM7, OC5, OC8, OC10, OC12), thereby improving interpretive clarity and the explanatory power of subsequent models.

Contrary to the initial hypothesis, the number of Agile practices adopted showed virtually no correlation with intra-team idea-exchange difficulty (R = –0.206) [4]. While richer practice portfolios are often assumed to foster collaboration, the data suggest that communication barriers depend more on how a few core mechanisms—daily stand-ups, sprint reviews, backlog grooming—are enacted than on sheer practice count. This finding aligns with Liu & Zhai's observation that the *quality* of Agile routines, rather than their quantity, drives effective information flow.

Practice diversity was likewise only weakly related to project duration (R = –0.215) and budget variance (R = 0.082). In other words, adding techniques such as pair programming or burn-down charts neither inflated costs nor reliably compressed schedules; instead, it sometimes accelerated delivery by streamlining hand-offs and reducing rework. A modest negative correlation with requirements-clarity issues on the client side further indicates that tools like user stories and sprint reviews improve transparency and goal alignment.

Assessment against the nine core Agile principles again showed strong scale reliability (Cronbach's α = 0.90). "Responding to change over following a plan" scored highest, whereas "customer collaboration over contract negotiation" lagged—evidence of lingering hesitance to engage external stakeholders openly. Overall, Agile and hybrid teams outperformed traditional ones on stakeholder satisfaction while matching them on schedule, budget, scope and quality. Regression models explained 21 % – 41 % of the variance in project success ($0.21 < R^2 < 0.41$), reinforcing earlier claims that Agile methods deliver measurable benefits in volatile environments [3].

The effects of Agile practices on conditions for project success are detailed in Table 2.

Table 2. Effects of Agile practices on conditions for project success [7]

	Agility		**Communication**				**Motivation**	**Transparency**	**Trust**	**Knowledge-Sharing**
	Extensiveness	**Efficiency**	**Internal**	**External**	**Direct**	**Indirect**				
AP1: Co-located team			-/++		+					+++
AP2: Customer involvement										+++
AP3: Self-organized team	-	+					-/+			
AP4: Continuous progress visualiza			+			+				+

	Agility		Communication				Motivation	Transparency	Trust	Knowledge-Sharing
	Extensiveness	Efficiency	Internal	External	Direct	Indirect				
tion										
AP5: Incremental releases				+	+					+
AP6: Minimal documentation										
AP7: Retrospectives			+		+	+	-/+	+	+	+
AP8: Value-prioritized requirements			-	+		+				
AP9: Continuous code integration			+							
AP10: Pair programming	+	N	-/+		+					
AP11: Sprints			-/+	+	+		-/+	+	-/+	+
AP12: Stand-up meetings	-	N	+		+		-/+	+	-/+	++

Project performance was evaluated based on a set of key success factors identified through a comprehensive literature review presented earlier in this study. Participants were asked to rate eight core elements of project success using a five-point Likert scale. The measurement instrument demonstrated high reliability, with a Cronbach's alpha of 0.9, indicating strong internal consistency and the appropriateness of the applied scale [5]. Mean scores across all eight parameters ranged from 3.3 to 3.6, with the highest ratings recorded for client satisfaction and adherence to budget constraints. These findings suggest that financial discipline and the creation of client value are perceived as the most critical indicators of success. Moreover, even when not all goals were fully achieved, respondents tended to view the final reports and the level of target attainment as the primary markers of positive project outcomes.

Conclusion

The evidence gathered confirms a clear positive relationship between the breadth of **Agile engineering practices** and higher success rates in software-development projects operating under uncertainty and rapid change. Significant correlations were observed between the use of core Agile routines and key outcome metrics—lower defect density, faster delivery cadence, heightened client satisfaction, and smoother team communication—demonstrating that short feedback loops and iterative reprioritisation translate into tangible technical and business gains.

At the same time, the comparatively modest link between practice adoption and organisational-culture variables suggests that Agile can yield benefits even where cultural maturity is uneven, provided that minimum conditions for transparency and shared ownership are met. The results therefore position Agile's principal strength not only in procedural flexibility but also in its disciplined, stakeholder-centric approach to building and evolving high-quality code.

Because implementation depth varied widely across contexts, further research should probe how specific combinations of practices and cultural enablers optimise technical performance in different organisational settings. Overall, the study reinforces the strategic value of Agile as a robust, empirically grounded framework for enhancing both engineering quality and project outcomes in the digital era.

References

1. Ali, H., Khan, M.Z., Rehman, U. ur. (2021). An Empirical Study on Adoption of Agile Project Management Methodology and Its Effect on Project Success with Moderating Role of Organizational Culture. European Journal of Social Impact and Circular Economy, 2(1), 75–99. https://doi.org/10.13135/2704-9906/5158 (accessed: 18.04.2025).
2. Ciric Lalic, D., Lalic, B., Delić, M., Gracanin, D., Stefanovic, D. (2022). How Project Management Approach Impacts Project Success? From Traditional to Agile. International Journal of Managing Projects in Business, 15(3), 494–521. https://doi.org/10.1108/IJMPB-04-2021-0108 (accessed: 11.04.2025).
3. Gemino, A., Horner Reich, B., Serrador, P.M. (2020). Agile, Traditional, and Hybrid Approaches to Project Success: Is Hybrid a Poor Second Choice? Project Management Journal, 52(2), 161–175. https://doi.org/10.1177/8756972820973082 (accessed: 16.04.2025).
4. Ghimire, D., Charters, S. (2022). The Impact of Agile Development Practices on Project Outcomes. Software, 1, 265–275. https://doi.org/10.3390/software1030012 (accessed: 09.04.2025).
5. Kanski, L., Budzynska, K., Chadam, J. (2023). The Impact of Identified Agility Components on Project Success—ICT Industry Perspective. PLoS ONE, 18(3), e0281936. https://doi.org/10.1371/journal.pone.0281936 (accessed: 17.04.2025).
6. Salman, A., Jaafar, M., Malik, S., Mohammad, D., Muhammad, S.A. (2020). An Empirical Investigation of the Impact of Communication and Employee Motivation on Project Success Using the Agile Framework and Its Effect on the Software Development Business. Business Perspectives and Research, 9(1), 46–61. https://doi.org/10.1177/2278533720902915 (accessed: 06.04.2025).
7. Sandstø, R., Reme-Ness, C. (2021). Agile Practices and Impacts on Project Success. Journal of Engineering, Project, and Production Management, 11(3), 255–262. https://doi.org/10.2478/jeppm-2021-0024 (accessed: 05.04.2025).
8. Wafa, R., Khan, M.Q., Malik, F., Abdusalomov, A.B., Cho, Y.I., Odarchenko, R. (2022). The Impact of Agile Methodology on Project Success, with a Moderating Role of Person's Job Fit in the IT Industry of Pakistan. Applied Sciences, 12(10698). https://doi.org/10.3390/app122110698 (accessed: 19.04.2025).
9. Dupret, K., Pultz, S. (2021). Beyond Agility: Flex to the Future. Pulse of the Profession / Project Management Institute. https://www.pmi.org/learning/library/beyond-agility-gymnastic-enterprises-12973 (accessed: 17.04.2025).