



Journal Website:
<http://sciencebring.com/index.php/ijasr>

Copyright: Original content from this work may be used under the terms of the creative commons attributes 4.0 licence.

 Research Article

Architectural and Organizational Dimensions of Legacy System Migration to Modern Cloud-Native Platforms: A Critical Examination of ASP.NET Core Evolution, Hybrid Modernization Strategies, and Socio-Technical Implications

Submission Date: November 01, 2025, **Accepted Date:** November 15, 2025,

Published Date: November 30, 2025

Samuel J. Lockwood

Department of Computer and Systems Sciences, Stockholm University, Sweden

ABSTRACT

Legacy system migration has emerged as one of the most complex and consequential challenges in contemporary information systems engineering. As organizations confront mounting pressures arising from digital transformation, regulatory compliance, cybersecurity threats, and escalating operational costs, the modernization of entrenched software systems has become unavoidable. This research article presents an extensive and theoretically grounded investigation into the architectural, methodological, and organizational dimensions of legacy system migration, with particular emphasis on the evolution from ASP.NET to ASP.NET Core as a representative case of platform-level transformation. Drawing on a broad and interdisciplinary body of literature, the study situates ASP.NET Core within wider discourses on cloud-native architectures, microservices, hybrid migration strategies, automation, and risk-based decision-making. Central to the analysis is the recognition that technical migration is inseparable from organizational culture, governance models, and human capital considerations. The article adopts a qualitative, interpretive research design grounded in comparative literature synthesis and conceptual analysis. The findings highlight that successful migration outcomes are contingent upon aligning architectural decisions with socio-technical realities, including legacy knowledge embedded in monolithic systems, institutional resistance to change, and evolving skill requirements. Particular attention is given to tooling ecosystems, implementation strategies, and incremental modernization pathways articulated in recent scholarship, including the evolution of ASP.NET Core as a modular, cross-platform, and performance-oriented framework (Valiveti, 2025). The discussion advances a multi-layered theoretical model that integrates architectural evolution, organizational readiness, and strategic governance, while also addressing limitations and unresolved debates in the field. By offering a deeply elaborated and critical



perspective, this article contributes to both academic scholarship and practitioner discourse on sustainable and resilient legacy system modernization.

KEYWORDS

Legacy system migration, ASP.NET Core, cloud-native architecture, digital transformation, hybrid modernization, organizational culture, software evolution

INTRODUCTION

Legacy information systems occupy a paradoxical position within modern organizations. On one hand, they represent accumulated institutional knowledge, proven reliability, and deeply embedded business logic that has evolved over decades of operational use. On the other hand, these systems increasingly constrain organizational agility, scalability, and innovation, particularly in an era defined by cloud computing, microservices, artificial intelligence, and data-driven decision-making (Chen & Xu, 2021). The tension between stability and adaptability has rendered legacy system migration not merely a technical undertaking, but a strategic and socio-technical challenge that intersects with organizational culture, governance, and risk management (Thomas & Powell, 2022).

Historically, legacy systems were designed under assumptions that differ fundamentally from those underpinning contemporary software architectures. Monolithic structures, tightly coupled components, and platform-specific dependencies were once rational responses to hardware limitations, cost structures, and development paradigms of earlier computing eras (Newman, 2019). However, these same characteristics now impede integration with cloud infrastructures, continuous deployment pipelines, and heterogeneous digital ecosystems. As

organizations pursue digital transformation initiatives, legacy systems increasingly emerge as bottlenecks that limit responsiveness to market changes and regulatory demands (Shivakumar, 2023).

Within this broader context, the evolution of web application frameworks provides a particularly illuminating lens through which to examine legacy modernization. The transition from ASP.NET to ASP.NET Core exemplifies a deliberate architectural rethinking aimed at addressing performance constraints, cross-platform compatibility, modularity, and cloud readiness. ASP.NET Core was not conceived as a mere incremental update, but as a re-architected framework designed to align with microservice-oriented and cloud-native paradigms (Valiveti, 2025). This transformation reflects wider industry shifts toward containerization, DevOps practices, and infrastructure abstraction, while simultaneously posing significant migration challenges for organizations with substantial investments in legacy ASP.NET applications.

The scholarly literature on legacy system migration has expanded significantly over the past two decades, encompassing technical strategies, methodological frameworks, and organizational perspectives. Early research emphasized code translation, reengineering, and system replacement as primary migration pathways. More



recent studies have highlighted hybrid approaches that combine incremental refactoring with selective redevelopment, often guided by agile or mixed-method methodologies (Zhao & Wang, 2020; Parker & Nguyen, 2022). Parallel strands of research have explored the role of automation, machine learning, and analytics in supporting migration decision-making and execution (Green & Miller, 2021; Harrison & Brooks, 2021).

Despite this growing body of work, several gaps remain. First, much of the literature treats architectural evolution and organizational dynamics as analytically separable, thereby underestimating their mutual influence. Second, while cloud migration strategies are widely discussed, fewer studies offer deep theoretical engagement with specific platform evolutions, such as the transition to ASP.NET Core, as exemplars of broader modernization trends (Liu & Zhang, 2020). Third, risk management is often framed in narrow technical terms, without sufficient attention to cultural resistance, skill obsolescence, and governance complexity (Robinson & Smith, 2021; Peterson & Reed, 2020).

This article seeks to address these gaps by offering an extensive, integrative analysis of legacy system migration that foregrounds the interplay between architecture, methodology, and organizational context. By situating the evolution of ASP.NET Core within a broader theoretical and historical narrative, the study aims to elucidate how platform-level changes both reflect and shape migration strategies. The central argument advanced is that legacy system modernization is best understood as an evolutionary process rather than a discrete event, requiring continuous

alignment between technical capabilities and organizational capacities.

The remainder of this article unfolds through a detailed methodological exposition, an interpretive presentation of results grounded in the literature, and an extended discussion that synthesizes competing scholarly viewpoints while advancing new theoretical insights. Throughout, the analysis maintains a critical stance, acknowledging both the promises and the limitations of contemporary migration approaches. By doing so, the article aspires to contribute to a more nuanced and sustainable understanding of legacy system migration in the age of cloud-native computing.

METHODOLOGY

The methodological orientation of this research is qualitative, interpretive, and theory-driven, reflecting the complex and socio-technical nature of legacy system migration. Rather than seeking to generate new empirical data through experimentation or surveys, the study adopts an extensive literature-based analytical approach designed to synthesize, critique, and extend existing scholarly knowledge. This methodological choice is consistent with prior research that emphasizes conceptual clarity and theoretical integration in fields characterized by rapid technological change and contextual variability (Chen & Xu, 2021).

The primary methodological framework employed is comparative literature synthesis. This involves systematically examining a diverse corpus of academic publications spanning software engineering, information systems, cloud computing, organizational studies, and digital transformation. The references considered include



peer-reviewed journal articles, conference proceedings, and authoritative monographs, each contributing distinct perspectives on legacy system migration. Particular emphasis is placed on works that address architectural evolution, hybrid modernization strategies, and organizational factors, thereby enabling a multi-dimensional analysis (Ponnusamy & Eswararaj, 2023).

A central methodological principle guiding this study is theoretical triangulation. By juxtaposing insights from different scholarly traditions, the analysis seeks to avoid reductionist interpretations that privilege either technical or organizational explanations in isolation. For example, architectural discussions of microservices and modularity are examined alongside studies of organizational culture and risk governance, allowing for a more holistic understanding of migration dynamics (Thomas & Powell, 2022; Newman, 2019).

The evolution of ASP.NET Core is treated as a focal case within this broader synthesis. Rather than functioning as an empirical case study in the traditional sense, it serves as an analytical anchor that illustrates key themes in legacy modernization, including platform re-architecture, tooling ecosystems, and implementation strategies. The selection of this focal case is justified by its prominence in both academic and practitioner discourse, as well as its explicit positioning as a response to legacy constraints within the .NET ecosystem (Valiveti, 2025).

Data extraction and analysis proceeded through iterative thematic coding of the selected literature. Key concepts such as modularization, automation, hybrid methodologies, risk management, and organizational readiness were identified and

progressively refined through repeated engagement with the texts. This process facilitated the identification of recurring patterns, points of contention, and underexplored dimensions across studies (Robinson & Smith, 2021).

The methodological limitations of this approach warrant careful consideration. First, reliance on secondary sources introduces the risk of interpretive bias, as the analysis is mediated through the perspectives and assumptions of prior authors. While this risk is mitigated through triangulation and critical comparison, it cannot be entirely eliminated (Green & Miller, 2021). Second, the absence of primary empirical data limits the ability to make context-specific claims about migration outcomes in particular organizational settings. However, the objective of this study is not empirical generalization, but theoretical enrichment and conceptual integration.

Another limitation concerns the temporal dynamics of technological evolution. Given the rapid pace of change in cloud platforms, frameworks, and tooling ecosystems, some technical details discussed in the literature may become outdated. To address this, the analysis emphasizes underlying principles and architectural patterns rather than transient implementation specifics (Liu & Zhang, 2020). The inclusion of recent conference research on ASP.NET Core evolution further enhances the temporal relevance of the study (Valiveti, 2025).

Ethical considerations are minimal in this context, as the study does not involve human subjects or proprietary data. Nevertheless, scholarly rigor and integrity are upheld through careful citation practices and critical engagement with sources.

In sum, the methodology adopted in this research is deliberately expansive and interpretive, designed to capture the complexity of legacy system migration as a socio-technical phenomenon. By integrating architectural, methodological, and organizational perspectives, the study provides a robust foundation for the subsequent analysis of results and theoretical discussion.

RESULTS

The results of this literature-based analysis are presented as a set of interrelated thematic findings that collectively illuminate the multifaceted nature of legacy system migration. Rather than enumerating discrete outcomes, the results articulate patterns of understanding that emerge from sustained engagement with the scholarly corpus. Each thematic finding is grounded in existing research and reflects convergences and divergences across studies (Chen & Xu, 2021).

One prominent finding concerns the redefinition of legacy systems themselves. Contemporary scholarship increasingly rejects static definitions that equate legacy solely with technological obsolescence. Instead, legacy systems are conceptualized as socio-technical artifacts that embody historical design decisions, organizational routines, and accumulated domain knowledge (Ponnusamy & Eswararaj, 2023). This reconceptualization has significant implications for migration strategies, as it underscores the risks of wholesale replacement and the value of incremental evolution.

A second thematic finding relates to the centrality of architectural modularization in enabling sustainable migration. The literature consistently

emphasizes that decoupling monolithic systems into modular components or services is a prerequisite for cloud adoption, scalability, and continuous deployment (Newman, 2019). The evolution of ASP.NET Core exemplifies this shift, as its lightweight, modular architecture contrasts sharply with the more monolithic design of earlier ASP.NET frameworks (Valiveti, 2025). This architectural transformation is widely interpreted as both a technical enabler and a catalyst for broader organizational change.

Automation emerges as a third key theme, particularly in relation to code analysis, testing, and deployment. Studies highlight the potential of automation tools to reduce migration effort, improve consistency, and mitigate human error (Harrison & Brooks, 2021). At the same time, the literature cautions against overreliance on automation, noting that complex business logic and undocumented dependencies often resist automated transformation (Green & Miller, 2021). The results thus suggest a nuanced role for automation as a supportive, rather than substitutive, mechanism.

Hybrid migration methodologies constitute a fourth thematic finding. Rather than adhering strictly to either waterfall or agile paradigms, organizations increasingly adopt hybrid approaches that combine upfront planning with iterative development and feedback (Zhao & Wang, 2020; Parker & Nguyen, 2022). This methodological hybridity is reflected in platform transitions such as ASP.NET to ASP.NET Core, where phased migration and coexistence of old and new components are common (Valiveti, 2025).

Organizational culture and governance emerge as a fifth and particularly salient theme. Multiple



studies document that cultural resistance, skill gaps, and misaligned incentives frequently undermine technically sound migration initiatives (Thomas & Powell, 2022). The results indicate that successful migration is as much a function of leadership, communication, and learning as it is of architectural design. Risk management frameworks that incorporate cultural and human factors are therefore increasingly advocated (Robinson & Smith, 2021; Peterson & Reed, 2020).

Finally, the literature reveals persistent tensions and unresolved debates. While cloud-native architectures promise scalability and flexibility, concerns remain regarding security, compliance, and operational complexity, particularly in regulated sectors such as finance and healthcare (Blake & White, 2021). These concerns complicate migration decisions and reinforce the need for context-sensitive strategies (Liu & Zhang, 2020).

Collectively, these results underscore the inadequacy of simplistic or purely technical narratives of legacy system migration. Instead, they point toward an integrative understanding that recognizes migration as an ongoing, negotiated process shaped by architectural possibilities, organizational constraints, and strategic priorities.

DISCUSSION

The discussion section provides an extended theoretical interpretation of the results, situating them within broader scholarly debates and drawing out their implications for both research and practice. Central to this discussion is the argument that legacy system migration should be understood as an evolutionary transformation

rather than a discrete project with a clearly defined endpoint (Newman, 2019).

From an architectural perspective, the evolution of ASP.NET Core exemplifies a broader shift toward modular, platform-agnostic, and performance-optimized frameworks. Scholars have noted that this shift reflects changing assumptions about deployment environments, development workflows, and scalability requirements (Valiveti, 2025). By decoupling core functionalities and embracing open standards, ASP.NET Core aligns with microservice and container-based architectures that are increasingly dominant in cloud ecosystems (Singh & Aggarwal, 2023). However, the discussion must also acknowledge counter-arguments that question whether such architectures introduce new forms of complexity and operational overhead (Yarygina & Bagge, 2018).

Methodologically, the prevalence of hybrid migration approaches challenges traditional dichotomies between plan-driven and agile methodologies. The literature suggests that hybrid models are not merely pragmatic compromises, but reflective of the inherent uncertainty and path dependency associated with legacy systems (Zhao & Wang, 2020). This insight resonates with organizational theories that emphasize learning and adaptation in complex systems (Thomas & Powell, 2022). Nevertheless, critics caution that hybrid approaches may suffer from methodological ambiguity and governance challenges if not carefully managed (Parker & Nguyen, 2022).

Organizational culture occupies a particularly contested space in the discussion. While there is broad consensus that culture matters, scholars

differ in their prescriptions for cultural change. Some advocate for transformational leadership and top-down mandates, while others emphasize participatory approaches and incremental learning (Robinson & Smith, 2021). The evolution of ASP.NET Core, with its emphasis on developer experience and open-source collaboration, can be interpreted as an attempt to reshape not only technical practices but also professional identities within the .NET community (Valiveti, 2025).

Risk management represents another area of debate. Traditional risk frameworks often prioritize technical failure modes, such as data loss or system downtime. However, the literature reviewed here supports a more expansive conception of risk that includes cultural resistance, skill erosion, and strategic misalignment (Peterson & Reed, 2020). This broader perspective aligns with socio-technical theories that view systems as inseparable from their organizational contexts.

The discussion also highlights limitations in the existing literature. Despite increasing attention to cloud migration, relatively few studies offer longitudinal analyses that track migration outcomes over extended periods. This gap limits understanding of long-term sustainability and technical debt accumulation (Chen & Xu, 2021). Furthermore, much of the research remains concentrated in specific industries or regions, raising questions about generalizability (Blake & White, 2021).

Future research directions emerge naturally from these limitations. Longitudinal and comparative studies could provide deeper insights into how migration strategies evolve over time and across contexts. Greater engagement with interdisciplinary theories, including organizational

psychology and political economy, could further enrich understanding of migration dynamics (Shivakumar, 2023).

CONCLUSION

Legacy system migration stands at the intersection of technological innovation and organizational transformation. This article has argued that understanding this phenomenon requires moving beyond narrow technical perspectives to embrace a holistic, evolutionary view. Through an extensive synthesis of the literature and a focused examination of the evolution from ASP.NET to ASP.NET Core, the study has highlighted the central roles of architectural modularity, hybrid methodologies, automation, and organizational culture.

The findings suggest that successful migration is less about replacing old systems than about cultivating adaptive capacities that enable continuous evolution. Platform transformations such as ASP.NET Core exemplify both the opportunities and the challenges inherent in this process, offering valuable lessons for organizations navigating the complexities of digital transformation (Valiveti, 2025).

By advancing an integrative theoretical perspective, this article contributes to ongoing scholarly debates and provides a foundation for future research and practice. As digital ecosystems continue to evolve, the capacity to manage legacy systems thoughtfully and strategically will remain a defining competence for organizations across sectors.

REFERENCES

1. Saxena, D., Gupta, R., & Singh, A. K. (2021). A survey and comparative study on multi-cloud architectures: Emerging issues and challenges for cloud federation.
2. Blake, P., & White, G. (2021). Legacy system migration in financial institutions: A case study approach.
3. Thomas, K., & Powell, J. (2022). The impact of organizational culture on legacy system migration.
4. Green, J., & Miller, T. (2021). Improving legacy system migration with machine learning techniques.
5. Peterson, N., & Reed, F. (2020). Comparing risk management strategies for legacy system migration.
6. Newman, S. (2019). Monolith to microservices: Evolutionary patterns to transform your monolith.
7. Chen, Y., & Xu, Z. (2021). Understanding the challenges of legacy system migration: A review of best practices.
8. Harrison, R., & Brooks, D. (2021). Automation in legacy system migration: Challenges and opportunities.
9. Zhao, J., & Wang, T. (2020). Agile methodology in legacy system migration: A hybrid approach.
10. Liu, Y., & Zhang, Q. (2020). Cloud migration strategies for legacy systems: An overview of migration tools and techniques.
11. Robinson, P., & Smith, L. (2021). Risk-based decision making in legacy system migration.
12. Parker, S., & Nguyen, M. (2022). Hybrid approaches to legacy system modernization: Combining waterfall and agile models.
13. Ponnusamy, S., & Eswararaj, D. (2023). Navigating the modernization of legacy applications and data: Effective strategies and best practices.
14. Shivakumar, S. (2023). Elements of digital transformation.
15. Singh, A., & Aggarwal, A. (2023). Artificial intelligence-enabled microservice container orchestration to increase efficiency and scalability for high-volume transaction systems in cloud environments.
16. Yarygina, T., & Bagge, A. H. (2018). Overcoming security challenges in microservice architectures.