

Modernizing Legacy Applications with AI

Challenges, insights, and evolution
with Forward by TDWI

Forward

Organizations across industries are under increasing pressure to modernize their technology environments to support new innovations such as artificial intelligence (AI). Systems that once formed the backbone of operations are now seen as obstacles to agility, innovation, and customer engagement. Yet modernization—particularly for legacy applications—is complex. Many organizations worry about disruptions to core business processes, data integrity, and the loss of critical functionality accumulated over years, if not decades, of operation.

While AI is often a goal in the modernization effort, it is also emerging as a key component of the process. AI is known primarily for its role in data analysis, decision support, and customer interaction, but its impact is expanding into software engineering, IT operations, and application transformation. TDWI research reflects this broader trend: While only about 22% of organizations report using generative AI in production today, more than 65% are actively experimenting with it. Moreover, fewer than 10% of organizations indicate they are not engaging with generative AI at all—underscoring the growing momentum behind AI adoption, even in its emerging forms.

Beyond experimentation, organizations are increasingly relying on AI-powered tools to streamline and automate critical processes. Seventy-four percent of recent TDWI survey respondents believe it is important to use “intelligent” tools that incorporate AI and machine learning (ML) to assist with data management tasks such as profiling, transforming, and processing data. Additionally, 84% agree that automating data management processes is essential for achieving operational success. Although these findings are centered primarily on data infrastructure, they highlight a broader shift: organizations are starting to trust AI and automation to enhance reliability, efficiency, and scalability across their technology ecosystems.

Application modernization represents a natural evolution of this shift. Modernizing legacy applications often involves untangling complex codebases, refactoring outdated architectures, and ensuring that new systems maintain—or

improve upon—the performance, reliability, and compliance of their predecessors. The use of AI in this context can greatly enhance the transformation process. AI can assist not only in code conversion but also in documenting functionality, generating robust testing frameworks, identifying optimization opportunities, and streamlining deployment workflows. When thoughtfully applied, AI can reduce the risk and time required for modernization while helping organizations maintain control over quality and business outcomes.

Yet, successful modernization requires more than AI and automation alone. Organizations must strike a balance between utilizing AI and applying proven engineering and data management principles. They will need to keep a human in the loop for some tasks. Additionally, stakeholder trust must be earned through transparency, iterative validation, and a careful focus on preserving core business functionality throughout the transformation journey.

This paper explores how MongoDB approaches this balance. By moving beyond monolithic transformation models to modular, slice-based modernization strategies supported by dependency graphs and AI augmentation, MongoDB demonstrates how enterprises can achieve modernization goals while maintaining data integrity and operational continuity. Their emphasis on conducting realistic pilot programs, tailoring solutions to client needs, and applying rigorous data engineering practices reflects the evolving best practices for modernization in the AI era.



Introduction

Legacy modernization is essential for organizations seeking to optimize operations, improve scalability, and deliver superior user experiences. Despite the availability of advanced AI tools, including LLMs, there's still a considerable gap between perceived and actual capabilities in transforming complex legacy systems.

Objective

This white paper aims to:

- Identify common misconceptions in application modernization using AI.
- Highlight the challenges faced in deploying LLMs for code transformation.
- Present effective strategies and methodologies for integrating AI in the modernization process.
- Provide insights from MongoDB's application modernization approach.



Understanding the Challenge: The Myth of Effortless Transformation

Initial assumptions

Initial approaches to modernization often rest on the oversimplified assumption that LLMs can seamlessly transform entire code repositories into functional modern applications. This is hardly true considering the nascent state of AI solutions and the scale and complexity of real-world legacy systems.

Key limitations of LLMs

1. **Contextual limitations:** LLMs possess inherent context window limits, which impede their ability to process large-scale applications effectively. This limitation results in a lack of spatial awareness across different files, leading to inconsistencies when updating methods or classes.
2. **Assessment challenges:** Early experimentation with AI-driven transformations—where code repositories were passed as is to LLMs—resulted in high test failure rates. Such experiments underscored the necessity for manual troubleshooting and revealed the inadequacies of a one-size-fits-all AI application approach in modernizing large applications.



A Structured Methodology for Effective Transformation

Phased approach to modernization

A structured and comprehensive methodology is essential for effective application modernization. Instead of relying solely on AI, a combination of traditional practices and AI technologies can lead to more successful outcomes.

Key phases in modernization

1. **Scoping and requirements gathering:**

This phase involves detailed scoping exercises to map out the existing architectural landscape and delineate clear functional and technical requirements. Stakeholder collaboration is crucial, often necessitating multiple interactions to validate and refine understanding.

2. **Understanding the existing application:**

A thorough analysis of the existing application is essential, including documenting functional and technical requirements and obtaining stakeholder sign-off.

3. **Design and implementation:**

Following requirement finalization, the design phase incorporates LLMs to assist in generating initial drafts of requirement documents and design prototypes.

4. **Testing and verification:**

LLMs are instrumental in quality verification, generating tests, and optimizing code to align with modern standards.

Revising the transformation process

Through iterative refinement, MongoDB has developed a more effective approach to transformation, moving from initial monolithic attempts to a modular, slice-based approach using dependency graphs. This methodology allows for targeted functionality transformation, reducing errors, and enhancing efficiency.



Leveraging AI Beyond Coding

While code transformation is a vital aspect of modernization, AI's potential extends further to encompass documentation, testing, and optimization.

Innovative applications of LLMs

1. **Documentation automation:** Automating the creation of functional requirements through recorded walkthroughs minimizes human error and documentation time.
2. **Testing and optimization:** AI can generate comprehensive testing code and evaluate performance metrics, ensuring consistency prior to deployment.
3. **Agentic modernization:** The integration of AI in every aspect of the modernization lifecycle, from design to deployment, exemplifies a holistic approach that leverages technology for enhanced outcomes.

Building Trust and Confidence in AI-Driven Modernization

Despite the potential benefits of AI, stakeholders remain skeptical, particularly regarding the preservation of existing functionalities.

Confidence through pilot programs

MongoDB addresses these concerns through pilot programs, which provide a controlled environment for assessing the complexity and feasibility of modernization projects. These programs enable:

- **Realistic scoping:** Access to legacy codebases during pilots allows for a thorough evaluation of project complexity, identifying precise requirements and challenges.
- **Stakeholder buy-in:** By demonstrating tangible progress and identifying potential roadblocks, MongoDB builds trust and gains agreement on project scope and timelines.

Aligning with customer needs

Successful modernization depends on understanding each client's unique requirements, tech stack, and operational landscape. MongoDB tailors its AI solutions accordingly, enabling seamless integration with varying customer technology environments.



MongoDB's Unique Contribution

MongoDB's expertise in data engineering sets its modernization services apart. Emphasizing the safe and accurate transmission of data throughout the modernization process ensures seamless integration and compatibility—critical considerations in preserving legacy data integrity.

Data-focused innovation

- **Database tracers and analyzers:** Tools developed by MongoDB prioritize the consistent and accurate delivery of data, allowing modernized applications to interface seamlessly with pre-existing data structures and reporting functions without disruption.
- **Comprehensive piloting:** By conducting pilots at no cost to customers, MongoDB provides informative insights into the transformation journey, ensuring transparency and setting realistic expectations for project outcomes.





Conclusion

Modernizing legacy applications requires balancing AI potential with traditional methodologies. MongoDB's approach, combining innovative AI applications with our strong data engineering foundation, offers comprehensive solutions for even complex transformations.

Through our partnerships and by talking to customers, MongoDB continues to evolve our application modernization strategies, ensuring that organizations capitalize on the full potential of their data on a modern, dynamic data layer.

For more information about MongoDB's approach to modernization, [contact sales.](#)

