NTT DATA

Revitalizing Application Development
A proactive strategy to reduce technical debt

Modernizing your business isn't just about investing in what's new and innovative. It also involves investing in the reconciliation of the outdated. Organizations are constrained by outdated IT processes and tools that are inflexible, siloed, and inefficient. This legacy infrastructure not only impedes development cycles and delays time-to-market but contributes to a significant accumulation of technical debt.

At NTT DATA, we will help you tackle your technical debt. We use the latest GenAI and Agentic AI innovations and capabilities along with our in-house expertise to not just conquer your technical debt; but conquer it quickly. We'll handle what's holding your business back so you can propel it forward.

This research from Everest Group, supported by NTT DATA, offers a strategic framework to overcome these challenges. The paper advocates for a modern, platform-based approach that fosters collaboration between IT and business teams, harnesses the power of AI and automation to streamline workflows, and empowers non-technical users through AI-enabled solutions. By integrating advanced technology with process transformation, enterprises can achieve greater agility, effectively reduce technical debt, and secure a competitive edge in an increasingly dynamic market.

Rajeev Singh
Senior Vice President
Global Application Services
NTT DATA, Inc.

Drew Gregory
Offering Leader
Global Application Services
NTT DATA, Inc.

Revitalizing Application Development

A Proactive Strategy to Reduce Technical Debt

Contents

Introduction

Today's business landscape is evolving at an unprecedented pace, driven by rapid technology disruption. To sustain and thrive in this fast-changing environment, enterprises must build an agile and flexible digital ecosystem. More importantly, they must adopt a proactive rather than reactive approach to understanding client expectations and deploying applications and enhancements faster and more efficiently to remain competitive.

However, many enterprises are constrained by traditional IT processes and methodologies, leading to rigidity and inefficiencies within their IT estates, which, in turn, hinder innovation. These limitations result in inefficient workflows, challenges in integrating new technologies, and an increasing burden of technical debt. Technology laggards carry a legacy application burden of up to 50%, while even enterprises with modernized technology estates grapple with a burden of up to 30%.[1] Consequently, organizations face rising maintenance costs, fragmented architectures, cybersecurity risks, and extended time-to-market, making ongoing application management increasingly difficult. To overcome these hurdles, enterprises must adopt modern application development practices to enhance agility and systematically reduce their technical debt.

This report examines how traditional development approaches contribute to technical debt and highlights the need for enterprises to adopt a more flexible and agile approach. It highlights:

- Key challenges with traditional application development approaches
- The need for a modern application development approach
- A platform-based development approach
- Low-code, generative AI capabilities in modern application development platforms
- A roadmap to transition from traditional to modern application development

Buyers can use this report as a comprehensive guide to transition to modern application development and build a future-ready digital ecosystem, while avoiding potential pitfalls.

1 Everest Group's interaction with over 200 enterprises

The traditional application development process – the root cause of obsolescence

Technology alone has never been sufficient to drive meaningful transformation without corresponding updates to underlying processes. Even the most advanced innovations fail to deliver their intended impact when integrated into outdated operational frameworks. A historical parallel can be drawn from the industrial revolution of the 1920s when factories recognized electricity as a groundbreaking innovation. However, rather than reengineering their production systems, they merely retrofitted electric motors onto outdated machinery, expecting significant productivity improvements. This approach yielded minimal gains, leading factories to attribute the failure to the technology rather than inefficient workflows.

Today, enterprises face a similar challenge as they integrate AI and modern technologies into legacy systems without transforming their software development processes, assuming that adopting next-generation technologies alone will drive innovation. However, without rethinking and modernizing underlying processes, these efforts often result in inefficiencies, increased operational complexity, and limited scalability. Exhibit 1 illustrates how enterprises today mirror this pattern by retrofitting modern technologies onto legacy systems rather than redesigning their software development approach, ultimately restricting innovation's true potential. Enterprises that fail to align technology adoption with process transformation struggle to optimize resources, effectively scale operations, and adapt to evolving market demands.

Many enterprises, in an attempt to modernize, choose quick-fix solutions rather than addressing foundational inefficiencies. Instead of transforming their processes to align with new technologies, they often opt for SaaS-based point solutions due to their ease of deployment and short-term gains. However, this fragmented approach does not address underlying inefficiencies and often creates an illusion of modernization. As a result, enterprises experience increased complexity, system fragmentation, and outdated technologies, ultimately accumulating disguised technical debt.

Without modernized and well-defined processes, application management becomes increasingly complex, impeding change, exacerbating inefficiencies, driving up costs, and limiting scalability – ultimately inhibiting innovation.

Hence, enterprises must ensure that their processes evolve in parallel. Without this alignment, technology innovation's intended benefits remain unrealized.

Challenges with the traditional application development process

The challenges associated with traditional application development are twofold – challenges within existing processes and those that emerge from the prolonged use of this approach over time. Exhibit 1 illustrates these challenges across three key enterprise dimensions: technology, processes, and people.

Exhibit 1: Challenges associated with legacy IT ecosystems and examples of technical debt

Source: Everest Group (2025)

	Challenges	Impact on enterprises and incurred technical debt
Technology	Bolt-on automation	Expensive and obsolete applications
	Security vulnerabilities	Cybersecurity risks
	Complex integrations	Workload scalability
Processes	Slow development cycles	Extended time-to-market
	Higher risk of errors	Increased operational risks
	Bottlenecks in deployment	Inefficient resource utilization
People	Limited collaboration	Misalignment between different functions
	Resistance to change	Limited innovation
	Skill gap	Talent retention challenges

To overcome these challenges, enterprises must adopt a more agile and flexible approach that accelerates deployments, ensures scalability, fosters collaboration, and reduces technical debt while enabling continuous innovation.
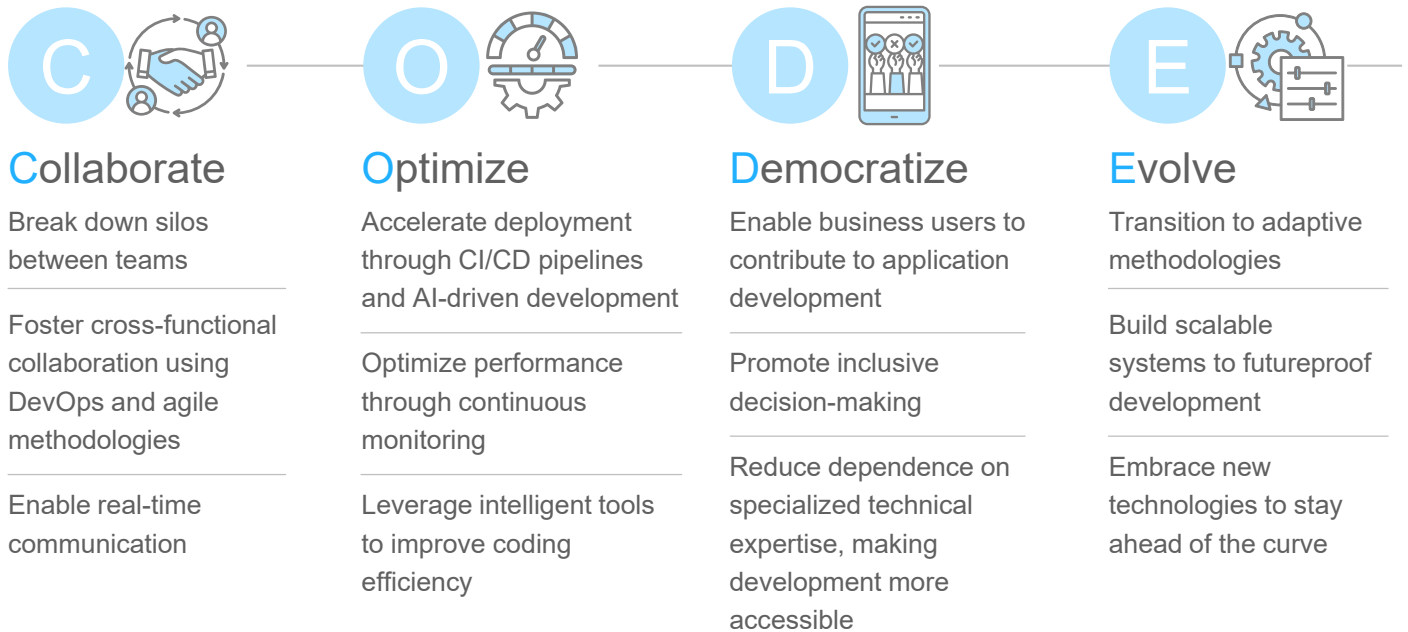
Emerging technologies' role in modern application development

Technology's rapid evolution has led to the rise of diverse application development tools and frameworks. These advances play a transformative role in reshaping development processes by bridging the gaps between people, processes, and technology. This not only streamlines application development but also enhances responsiveness to market and technology disruptions while improving applications' quality, scalability, and maintainability.

The key principles of modern application development are explained through the CODE framework illustrated in Exhibit 2.

Exhibit 2: Key principles of modern application development

Source: Everest Group (2025)

Collaborate

Break down silos between teams

Foster cross-functional collaboration using DevOps and agile methodologies

Enable real-time communication

Optimize

Accelerate deployment through CI/CD pipelines and AI-driven development

Optimize performance through continuous monitoring

Leverage intelligent tools to improve coding efficiency

Democratize

Enable business users to contribute to application development

Promote inclusive decision-making

Reduce dependence on specialized technical expertise, making development more accessible

Evolve

Transition to adaptive methodologies

Build scalable systems to futureproof development

Embrace new technologies to stay ahead of the curve

The CODE framework for modern application development

The CODE framework highlights modern application development's holistic aspects, enabling enterprises to accelerate innovation while staying relevant and competitive. This approach covers how modern application development enables collaboration, optimized efficiency, inclusive development, and continuous evolution to drive long-term business success. Let us look at the framework more closely:

- **Collaborate:** Modern application development facilitates seamless collaboration across teams, enhancing efficiency and driving innovation. By leveraging methodologies such as DevOps and agile, enterprises can eliminate silos, foster cross-functional alignment, and improve responsiveness to evolving business and market demands. It also promotes real-time communication and shared accountability, enabling faster iteration cycles, fostering continuous innovation, and reducing operational inefficiencies. Enterprise application development platforms have emerged as this transformation's key enablers, providing integrated environments that enhance collaboration, optimize resource utilization, and maximize technology investments. These platforms help establish a scalable and adaptive development ecosystem, ensuring enterprises can continuously evolve and respond to dynamic market needs

- **Optimize:** Modern application development primarily focuses on optimizing efforts across key activities by enabling enterprise-wide automation, which enhances performance, efficiency, and time-to-market. Continuous Integration and Continuous Deployment (CI/CD) pipelines, AI-driven development, and continuous monitoring allow enterprises to achieve faster deployments, improve code quality, and enhance system resilience

- **Democratize:** Modern technologies, such as low-code/no-code platforms, cloud-native frameworks, and AI-driven development, democratize application development by bringing business users closer to the process. These innovations enable non-technical users to contribute meaningfully to application development

- **Evolve:** Modern application development is not a one-time transformation but a continuous innovation process. Enterprises must be able to transition to adaptive methodologies and embrace emerging technologies to stay ahead of the curve. Modern application development enables future-proofing development through iterative improvements and continuous technology evolution
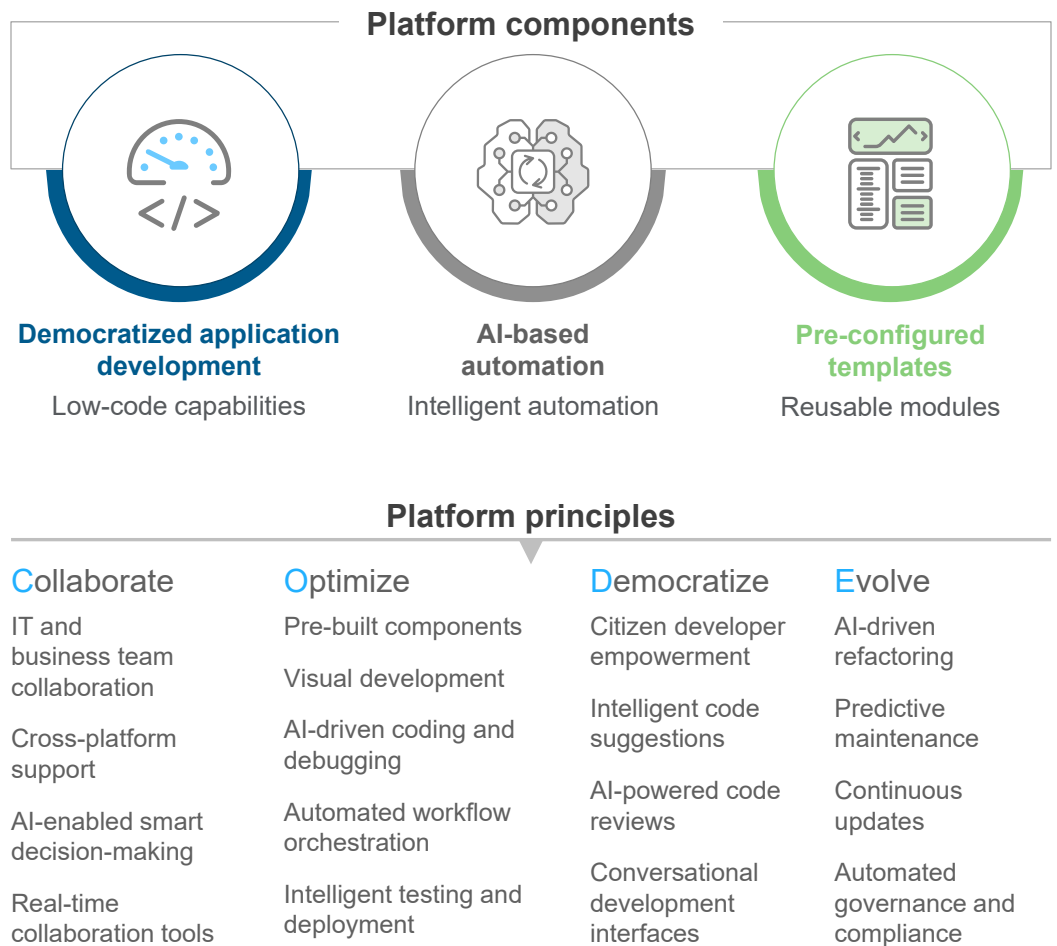
These principles offer key considerations for enterprises to reduce overall technical debt by positioning IT as a strategic business enabler.

A platform-based development approach

Today's fast-paced digital landscape demands an iterative and collaborative software development approach, rendering the traditional siloed Software Development Life Cycle (SDLC) ineffective. To meet these evolving requirements, a platform-based development approach is emerging as an efficient way to build and deploy applications. Exhibit 3 depicts the key principles of A platform-based development approach.

Exhibit 3: Application development platforms' key components and principles
Source: Everest Group (2025)

Platform components

Democratized application development
Low-code capabilities

AI-based automation
Intelligent automation

Pre-configured templates
Reusable modules

Platform principles

Collaborate	**O**ptimize	**D**emocratize	**E**volve
IT and business team collaboration	Pre-built components	Citizen developer empowerment	AI-driven refactoring
Cross-platform support	Visual development	Intelligent code suggestions	Predictive maintenance
AI-enabled smart decision-making	AI-driven coding and debugging	AI-powered code reviews	Continuous updates
Real-time collaboration tools	Automated workflow orchestration	Conversational development interfaces	Automated governance and compliance
	Intelligent testing and deployment		

A platform-based application development model focuses on creating scalable, reusable, and market-driven solutions that serve a broad user base rather than a single client or project. Iterative development and user feedback loops ensure that applications remain adaptable, competitive, and aligned with evolving market demands. Below are the key features of A platform-based application development approach aligned with the CODE framework:

Collaborate

- **IT and business team collaboration:** facilitates cross-functional teamwork, ensuring development aligns with business objectives
- **Cross-platform support:** enables applications to be developed and deployed seamlessly across multiple environments, enhancing flexibility
- **AI-enabled smart decision-making:** leverages AI-powered insights for faster, data-driven decisions, improving efficiency
- **Real-time collaboration tools:** allows multiple teams to work simultaneously on application design, testing, and deployment, thereby reducing silos

Optimize

- **Pre-built components:** reduces development time by leveraging reusable templates, APIs, and modular code
- **Visual development:** simplifies creating applications through drag-and-drop interfaces and pre-configured workflows
- **AI-driven coding and debugging:** enhances productivity with AI-assisted code generation, debugging, and optimization
- **Automated workflow orchestration:** reduces manual intervention by enabling AI-powered automation of repetitive tasks
- **Intelligent testing and deployment:** AI-driven testing tools identify bugs, optimize performance, and ensure seamless releases

Democratize

- **Citizen developer empowerment:** allows non-technical users to participate in application development through intuitive platforms
- **Intelligent code suggestions:** AI-powered tools assist developers by suggesting optimizations and auto-generating code
- **AI-powered code reviews:** enhances code quality with automated reviews, detecting inefficiencies and security risks
- **Conversational development interfaces:** enables creating applications using natural language inputs and AI-assisted design recommendations

Evolve

- **AI-driven refactoring:** identifies and restructures inefficient code to reduce technical debt and improve performance
- **Predictive maintenance:** uses analytics to anticipate failures and proactively optimize system performance

- **Continuous updates:** ensures applications remain up-to-date, secure, and aligned with evolving business needs
- **Automated governance and compliance:** AI-powered monitoring enforces regulatory compliance and security best practices

Low-code capabilities for faster and accessible development

Low-code platforms emerged as strong game changers, bridging the gap between unrelenting business needs and IT capacity. These capabilities simplify coding and accelerate development while minimizing technical debt risk. Enterprises can reduce development cycles, optimize resources, and increase agility without extensive coding expertise. Below are low-code platforms' key features:

- **Visual development environment:** Drag-and-drop interfaces allow rapid application design and development without extensive coding
- **Pre-built components and templates:** Reusable modules, APIs, and workflows streamline application development and reduce effort
- **Automation and AI integration:** AI-driven automation optimizes processes, enhances decision-making, and improves development efficiency
- **Cross-platform compatibility:** The applications built can be deployed seamlessly across web, mobile, and desktop environments
- **Seamless integrations:** Built-in connectors enable integration with databases, enterprise applications, and third-party services
- **Collaboration and governance:** They enable real-time collaboration between IT and business users while maintaining security and compliance controls
- **Security and compliance:** Built-in security measures, access controls, and compliance frameworks ensure application safety and regulatory adherence
- **Scalability and flexibility:** They support modular architectures, allowing businesses to extend applications with custom code, as needed
- **Citizen developer enablement:** Low-code platforms empower non-technical users to create applications through intuitive, low-code interfaces
- **CI/CD and DevOps support:** They enable automated deployment, testing, and continuous integration for faster release cycles

Owing to these advantages, platforms with low-code capabilities have seen significant adoption in recent years.

Generative AI, agentic AI, and intelligent automation in application development

Generative AI and agentic AI are revolutionizing application development by automating key processes, enhancing collaboration, and making development more accessible to a broader user range. With the ability to automate multiple phases across the SDLC, AI enables developers to create, test, and deploy applications faster, improving efficiency while minimizing technical debt. Below are some ways in which AI is reshaping application development.

Automating code generation
AI-powered tools assist developers by generating, optimizing, and suggesting code based on natural language inputs or existing patterns.

Intelligent workflow orchestration:
AI-driven agents manage end-to-end SDLC workflows, automating dependencies, prioritizing tasks, and ensuring seamless CI/CD pipelines

Optimizing application performance
AI-powered insights help predict system failures, recommend performance enhancements, and automate updates.

Democratizing development
Enables non-technical users to create applications using AI-driven tools, reducing reliance on IT teams and fostering innovation.

Autonomous debugging and issue resolution
AI agents proactively detect, diagnose, and resolve issues in applications without human intervention, reducing downtime and improving reliability

Refactoring code
Identifies inefficiencies in codebases, recommends optimizations, and automates refactoring to improve maintainability and scalability.

Accelerating SDLC
Automates testing, debugging, and deployment, reducing manual effort and expediting time-to-market.

By integrating generative AI into modern application development, enterprises can accelerate software delivery significantly. As enterprises continue to embrace automation, AI is set to become the application development process's cornerstone, enabling enterprises to remain agile and competitive in an evolving technology landscape.

Navigating the shift to a modern application development approach

Modern application development requires enterprises to move beyond rigid, legacy frameworks and adopt agile, scalable, and AI-driven methodologies. This transition involves rethinking development processes while addressing key challenges to ensure seamless adoption, minimize technical debt, and enable continuous innovation with faster time-to-market.

Key challenges in transitioning to modern application development

Transitioning to a modern application development approach is a complex process that impacts various business aspects. It is essential to adopt a structured approach to facilitate a seamless transition while addressing key challenges. Below are key challenges that hinder the transition:
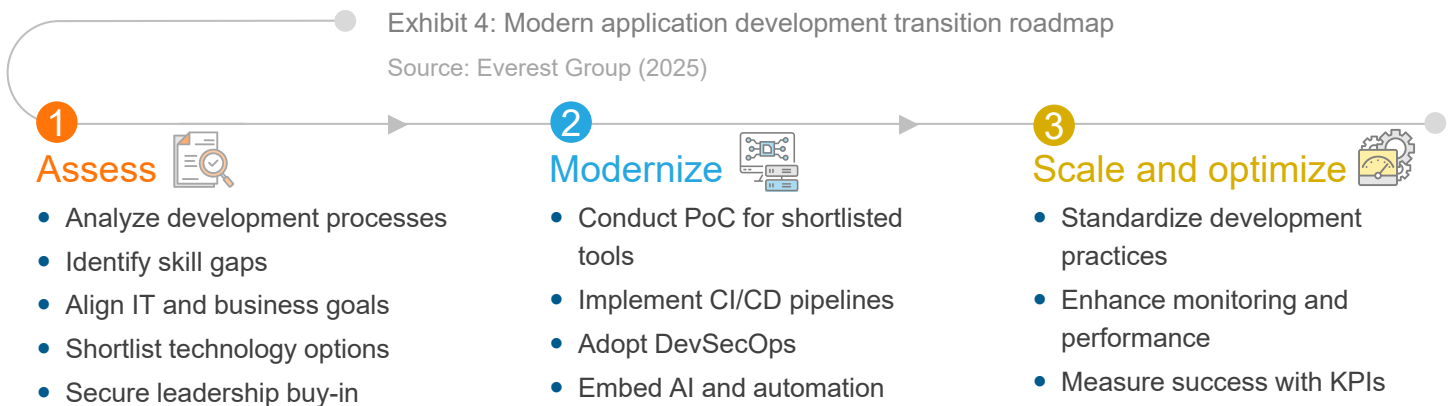
- **Resistance to change and cultural shift:** Resistance to change is the most significant challenge when enterprises transition to modern application development. The fear of uncertainty and business disruption often create misalignment among employees, leading to resistance. Enterprises can overcome this by fostering a culture of collaboration by involving teams early in the transition, ensuring they understand modern methodologies' benefits. Also, effective change management strategies, such as leadership sponsorship and internal advocacy, can help enterprises drive engagement and reduce resistance

- **Security and compliance risks:** The rapid iterations in agile and DevOps models can introduce security vulnerabilities into the system. To mitigate this risk, enterprises must integrate DevSecOps practices to embed security into every development phase rather than treating it as a bolt-on feature. Also, leveraging AI-based vulnerability assessment tools can enhance threat detection capabilities to keep the system resilient

- **Skills gaps and upskilling challenges:** While modern application development promotes democratization, adopting a modern technology stack requires specialized IT expertise to build an efficient and scalable ecosystem. The lack of necessary skills can extend the transition period and result in inefficiencies, leading to increased costs. To address this, enterprises should conduct a comprehensive skills assessment to identify gaps in modern technologies. Investing in structured training programs, certifications, and hands-on learning through hackathons can accelerate skill development and facilitate a seamless transition

- **Tooling and technology complexity:** With the vast number of available tools and technologies, selecting the right ones presents a key challenge and can, in some cases, lead to fragmented workflows and inefficiencies. To mitigate this, enterprises should establish a standardized technology stack aligned with business and development objectives to ensure consistency and efficiency across teams. Conducting Proof-of-Concept (PoC) pilots before full-scale adoption enables a thorough evaluation of tool effectiveness and compatibility

- **Lack of clear roadmap and governance:** The absence of a clearly defined roadmap and governance can cause misalignment between business and IT teams, leading to inconsistent modernization efforts. To address this concern, enterprises should establish a structured modernization roadmap with well-defined milestones and KPIs. Implementing governance frameworks that align development with business priorities enhances coordination and accountability

Enterprises must address these challenges as they transition to a modern application development technology stack.

Next, we detail a strategic roadmap to facilitate a seamless and effective transformation.

Transition roadmap

Transitioning to modern application development requires a structured and phased approach. Our roadmap provides clear, practical guidance to enterprises to enhance scalability, agility, and innovation, while ensuring security and governance. Exhibit 4 shows the key roadmap phases to transition to modern application development.

Exhibit 4: Modern application development transition roadmap
Source: Everest Group (2025)

1 Assess
- Analyze development processes
- Identify skill gaps
- Align IT and business goals
- Shortlist technology options
- Secure leadership buy-in

2 Modernize
- Conduct PoC for shortlisted tools
- Implement CI/CD pipelines
- Adopt DevSecOps
- Embed AI and automation

3 Scale and optimize
- Standardize development practices
- Enhance monitoring and performance
- Measure success with KPIs

We deep dive into each of these phases below.

Assess
- **Analyze development processes:** Comprehensively review existing workflows, technology stacks, and architectural inefficiencies to identify improvement areas
- **Identify skill gaps:** Assess teams' technical readiness for agile, DevOps, cloud, and automation, highlighting areas that require upskilling
- **Align IT and business goals:** Define key objectives, focusing on improving speed, security, scalability, and cost efficiency, along with relevant KPIs to measure success
- **Shortlist technology options:** Evaluate and shortlist tools, platforms, and frameworks for cloud, automation, CI/CD, AI, and security
- **Secure leadership buy-in:** Gain executive sponsorship and cross-functional alignment to ensure a phased and well-supported execution strategy

Modernize
- **Conduct PoC for shortlisted tools:** Run pilot projects to evaluate shortlisted tools' effectiveness and adopt best-fit solutions
- **Implement CI/CD pipelines:** Automate software integration, testing, and deployment to reduce errors and improve release efficiency
- **Adopt DevSecOps:** Embed security controls and compliance measures throughout the development life cycle to mitigate risks
- **Embed AI and automation:** Implement AI-driven testing, monitoring, and workflow automation to optimize efficiency and improve decision-making

Scale and optimize

- **Standardize development practices:** Establish coding standards, governance policies, and best practices to ensure consistency and compliance
- **Enhance monitoring and performance:** Deploy real-time analytics, observability tools, and AI-powered anomaly detection to improve system health and reliability
- **Measure success with KPIs:** Track and evaluate progress using deployment frequency, lead time for changes, mean time to recovery, defect rates, security compliance scores, and infrastructure cost efficiency to ensure sustained improvement and value realization

While the roadmap may seem straightforward, the transition can become complex if not effectively executed. Engaging a strategic partner with experience across multiple clients and large-scale implementations facilitates a consultative approach to the transition. Such partners also bring industry partnerships and proprietary solutions that enable a cost-effective modernization process. Selecting the right strategic partner ensures a seamless and successful transition.

Conclusion

As traditional approaches make it difficult for enterprises to adapt to evolving business demands, transitioning to modern application development is no longer optional but necessary. With the modernization market surging to over US$160 billion in the last decade, the demand for upgrading legacy systems and processes has never been more urgent.[2] To remain competitive, enterprises must strategically plan their shift toward an iterative, AI-enabled, and data-driven development model. This transformation not only helps address technical debt concerns but also enhances agility, enabling enterprises to respond quickly to market changes and customer expectations.

However, this transition requires a thoughtful and well-structured approach rather than a one-size-fits-all strategy. Enterprises must begin by assessing their technology ecosystems, development workflows, and skill gaps to ensure that modernization efforts align with their business objectives. Simply adopting modern tools without a strategic foundation could result in inefficiencies, failing to create the differentiation and competitive edge needed in today's digital landscape.

Enterprises that resist change will face mounting technical debt, declining agility, and an inability to meet customer expectations. By proactively adopting a modern, intelligent application development approach, enterprises can future-proof their operations and drive continuous innovation in an increasingly dynamic and competitive landscape. The question is no longer if enterprises should transform but how fast they can shift before they are left behind.

Everest Group®
With you on the journey

Everest Group is a leading research firm helping business leaders make confident decisions. We guide clients through today's market challenges and strengthen their strategies by applying contextualized problem-solving to their unique situations. This drives maximized operational and financial performance and transformative experiences. Our deep expertise and tenacious research focused on technology, business processes, and engineering through the lenses of talent, sustainability, and sourcing delivers precise and action-oriented guidance. Find further details and in-depth content at www.everestgrp.com.

For more information about this topic please contact the author(s):

Alisha Mittal, Vice President
alisha.mittal@everestgrp.com

Lalith Kumar, Practice Director
lalith.kumar@everestgrp.com

Notice and Disclaimers