JULY 2024

The State of DevOps: Accelerating Software Development With Generative AI

Jon Brown, Senior Analyst

Abstract:

The ever-increasing importance of software as a business driver and the relentless pressure to deliver high-quality applications faster is driving a shift towards leveraging generative AI in application development. This white paper explores the potential of generative AI to address these challenges and empower engineering leaders. It finds that organizations are rapidly adopting cloud-native development to accelerate application delivery. Generative AI promises to significantly improve developer productivity, code quality, and testing efficiency.

While leaders are enthusiastic about AI, skills gaps, implementation risks, and inexperience with the vast array of tools that are becoming available remain top concerns. To that end, this paper also outlines a strategic approach to AI adoption, including identifying high-impact use cases like code generation, documentation creation, and test case automation, to name a few areas that engineering leaders in the know can address today.

By strategically embracing AI, engineering leaders can unlock a competitive advantage and pave the way for a future of faster, more secure, and innovative software development while minimizing risks by following best practices.

Why Engineering Leaders Want Generative AI in Application Development Now

Two factors are coming together that make this moment critical to the present and future of application development: the increasing importance of software as a revenue driver for business and the internal desire to increase productivity and software quality. The arrival of AI promises to be an enabler and tireless extension of human capabilities for coding, quality assurance testing, and problem/defect triage, helping increase productivity and software quality.

The Increasing Importance of Software to Business

Watts S. Humphrey, widely regarded as the father of software quality, opined that "every business is a software business." This decades-old assertion is becoming increasingly true in a world where internally developed or custom software now generates more than 25% of total revenue for 59% of organizations, according to a recent survey conducted by TechTarget's Enterprise Strategy Group.[1]

Against that backdrop, 91% of organizations intend to increase cloud-native application development over the next 12-18 months.[2] A big part of that increase is directed at AI investments designed to enhance the creation and delivery of custom and in-house-written software to serve customers, employees, and partners.

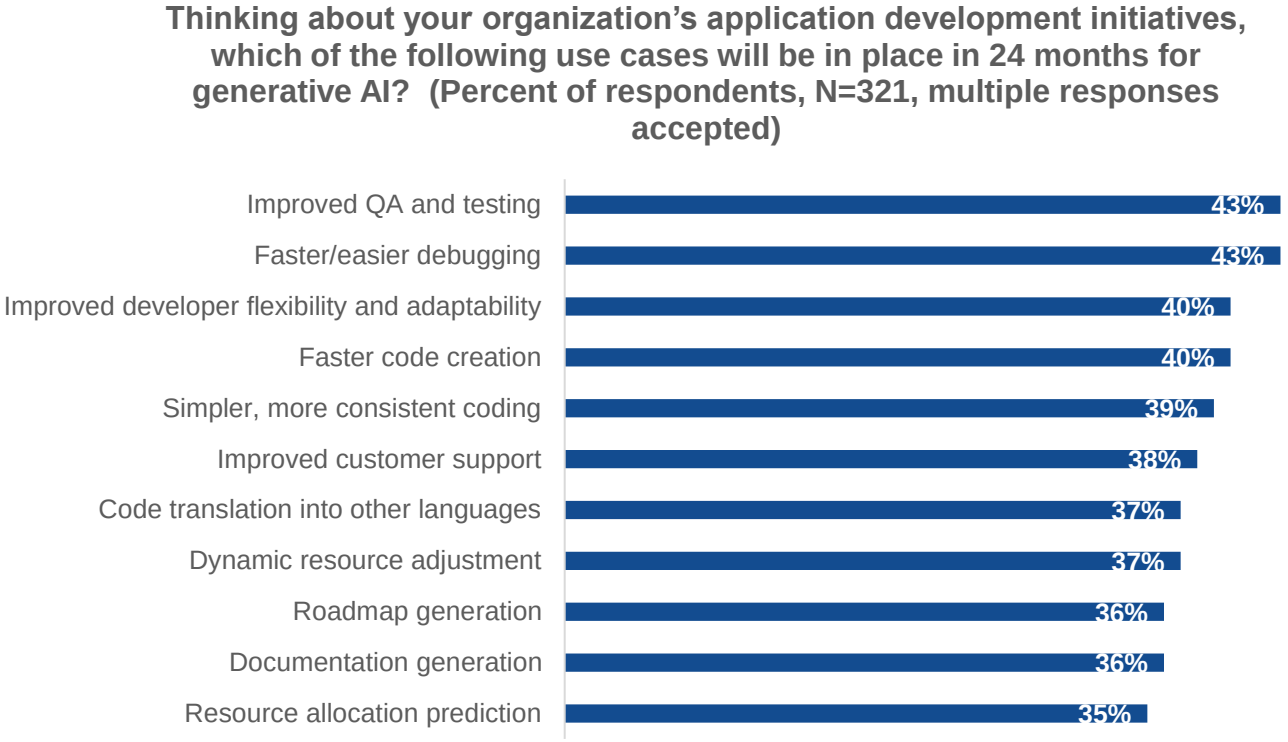
[1] Source: Enterprise Strategy Group Research Report, *Distributed Cloud Series: Understanding Buyer and Influencer Personas*, December 2023.
[2] Source: Enterprise Strategy Group Complete Survey Results, *Distributed Cloud Series: The Mainstreaming of Cloud-native Apps and Methodologies*, May 2023.

The Search for Acceleration: Delivering High-value, High-quality, and Secure Software Faster

The pressure to achieve "more, faster" is a constant across all business functions, and software development is no exception. Another recent Enterprise Strategy Group study underscores this urgency: 72% of respondents reported a significant increase in the need to accelerate application and infrastructure deployment over the past three years. They're achieving this through automation and process streamlining, with 32% reporting deployment speeds 50%-100% faster than just three years ago.[3] In the same report, it is clear that this acceleration is continuous and that leaders were looking to AI and automation to help them continue improving metrics for code quality, deployment speed, and innovation.

Yet another report reveals that top planned use cases for generative AI in application development include improving QA and testing (cited by 43%), enabling faster code creation (40%), and producing more straightforward and consistent code (39%; see Figure 1).[4] If implemented correctly, these planned use cases will improve software quality and delivery speed and help organizations meet their acceleration and productivity goals.

Figure 1. How Organizations Intend to Benefit From Generative AI investments in Application Development



Source: Enterprise Strategy Group, a division of TechTarget, Inc.

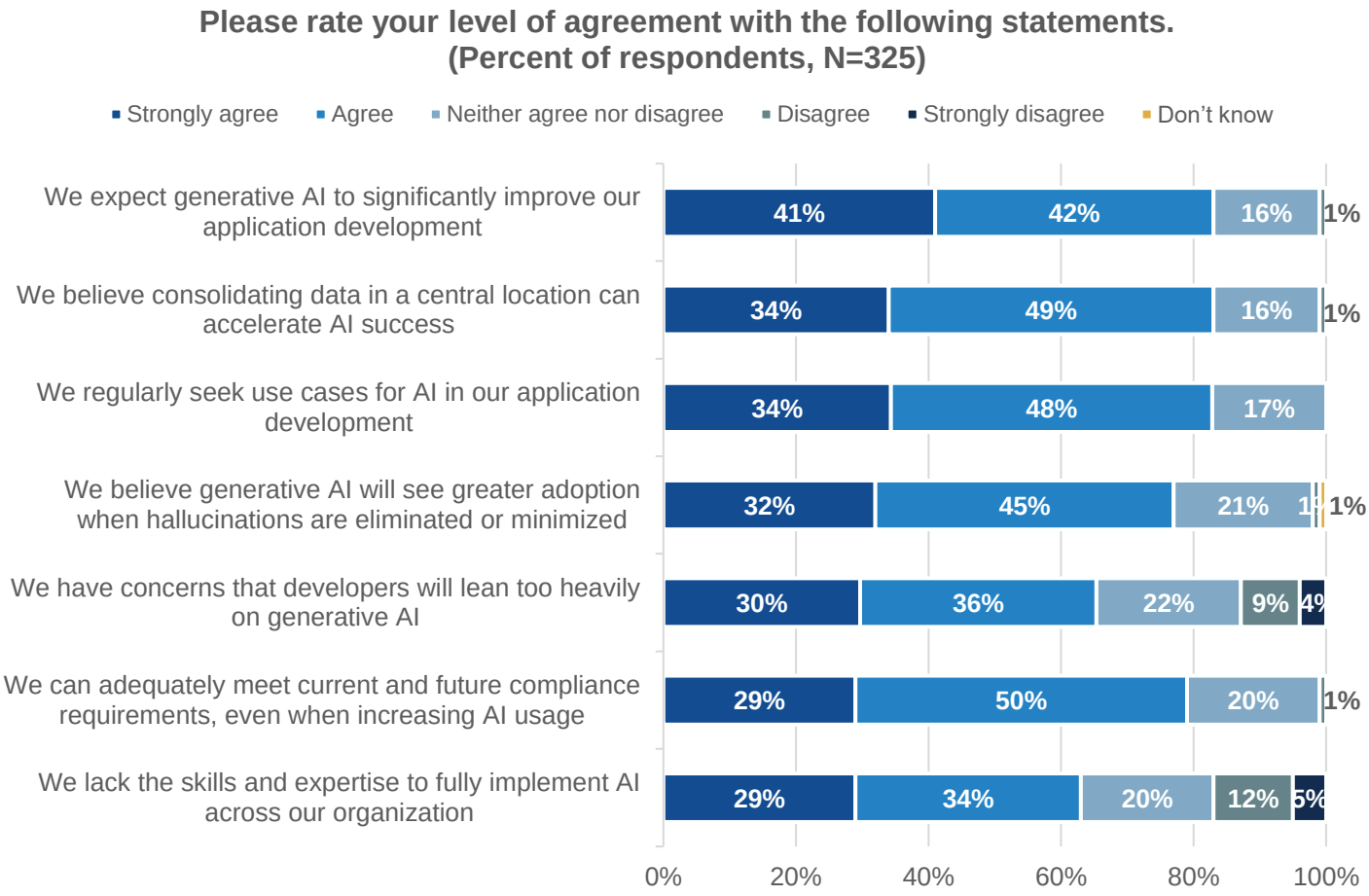
AI: A Promising Enabler, But a Lack of Skills and Expertise Stand in the Way

Anticipation is high for AI in application development. When asked about their expectations for generative AI in application development, 83% of respondents said they expect generative AI to improve application development

[3] Source: Enterprise Strategy Group Research Report, *Distributed Cloud Series: The State of Infrastructure Modernization Across the Distributed Cloud*, November 2023.
[4] Source: Enterprise Strategy Group Complete Survey Results, *Code Transformed: Tracking the Impact of Generative AI on Application Development*, April 2024.

significantly (see Figure 2). However, Figure 2 also shows the majority (63%) of respondents agreed that they need more skills and expertise to implement AI fully across the organization.[5] Organizations will need additional skills and expertise to meet the challenge and seize the opportunity that generative AI presents for software development organizations.

Figure 2. High Hopes for AI Amid Concerns Around Skills Gaps and Compliance



Source: Enterprise Strategy Group, a division of TechTarget, Inc.

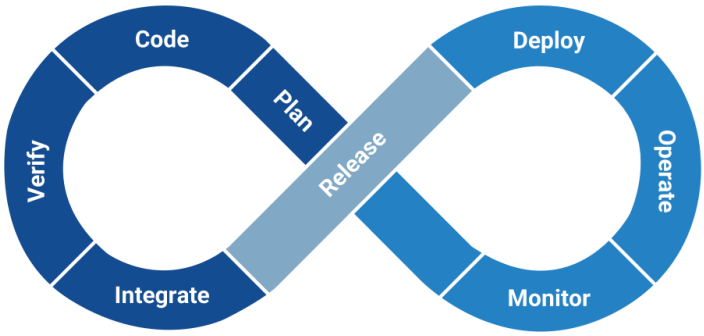
The high stakes of AI implementation in software development—including the potential for soaring productivity and the risk of unforeseen but high-profile failure—place engineering leaders at a critical juncture. Leaders must navigate the demands for speed, efficiency, security, and quality that AI promises alongside potential perils and missteps of adopting a new, paradigm-shifting technology.

Unlocking the Potential of AI in Application Development

AI can add value across the software application lifecycle, from value creation in the form of coding, planning, and deployment through value realization stages, where AI can benefit IT operations and support teams. To get the most benefit, organizations should follow a process, illustrated in Figure 3, that maximizes benefits while minimizing risk.

[5] Ibid.

Figure 3. Opportunities to Leverage AI Across the SDLC



Code
AI for developers
- Code / Project / Repo Summarization
- Predictive Code Generation
- Interactive Code Generation
- Test Generation
- Refactoring
- Debugging
- Documentation generation

Plan
AI for Product Managers
- Summarize planning meeting notes
- Agile planning
- Generate acceptance tests from stories
- Summarize issue discussions

AI for Architecture/Design
- Generate threat models from designs
- Convert visuals to infra / code

Deploy
AI for Release Managers
- Continuous deployment
- Infrastructure provisioning
- Container management
- Environment management

Operate
AI for Operators
- Adaptive optimization
- Pattern recognition, anomaly detection
- Continuously detect and repair config drift
- Incident response

Verify
AI for developers
- Test Generation
- Code Analysis
- Synthetic test data generation

AI for Security
- Detect, explain, remediate vulnerabilities
- Secrets management

Integrate
AI for developers
- Package Management
- Container Management
- SBOM
- Build automation

Release
AI for Release Managers
- Change impact analysis
- Configuration management

Monitor
AI for On-Call
- Incident response
- Natural Language Querying
- Aid root-cause analysis/correlate changes to alerts / summarize log data
- Reduce alert-fatigue

Source: Enterprise Strategy Group, a division of TechTarget, Inc.

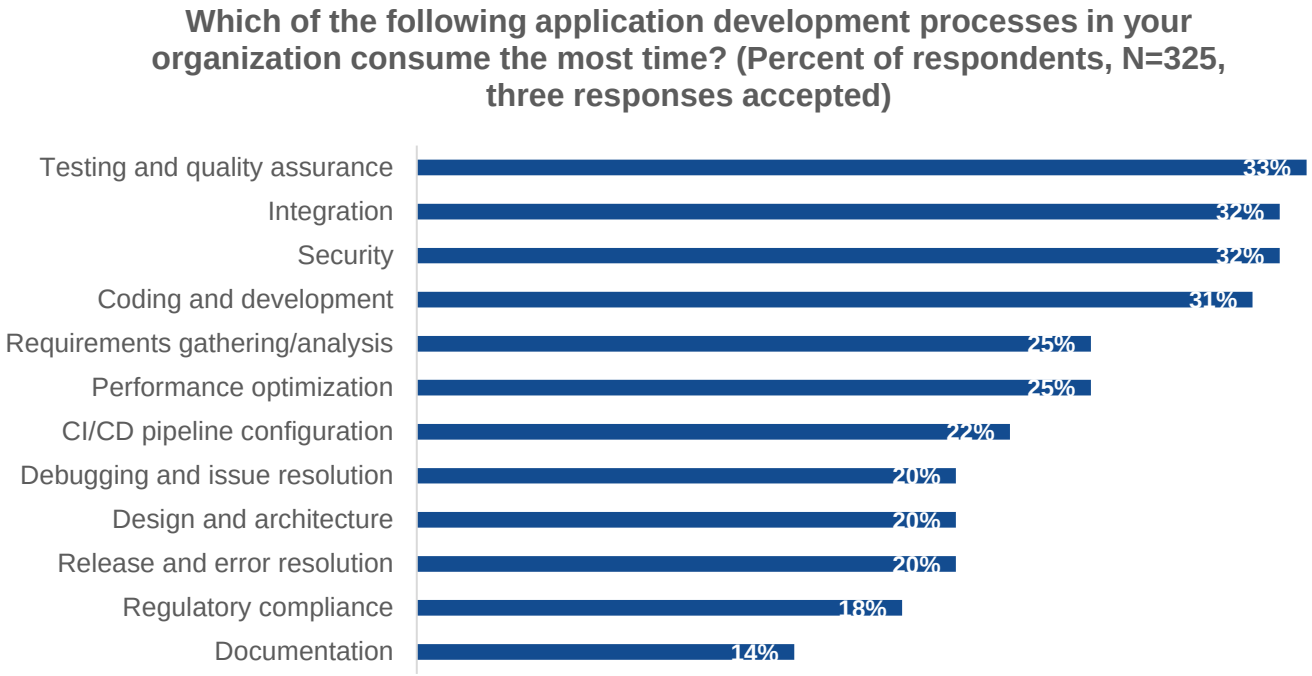
By following a defined methodology, organizations can benefit from others' experience and avoid common pitfalls associated with changing the way application developers create, modify, and maintain code. The following paragraphs give a high-level overview of critical considerations at each stage of the journey.

Define Your Goals and Identify Early Use Cases

Look for bottlenecks in the application development process. While actual coding and developer AI assistants or copilots are very high profile and currently get the most attention in the media, the activities most often cited as consuming the most time are testing and QA, integration, and security (see Figure 4).[6] This means that all of the phases are important, and it also means that currently no single tool can deliver on all facets of AI in the software development lifecycle (SDLC). Assistants are early successes but expect to find that each stage of the SDLC may have a host of competing tools to address that particular phase.

[6] Ibid.

Figure 4. Top Reported Bottlenecks in Application Development



Source: Enterprise Strategy Group, a division of TechTarget, Inc.

Set Goals

Clearly define your AI goals based on a thorough analysis. Are you aiming to boost developer productivity by automating repetitive tasks? Perhaps you seek to minimize bugs through AI-powered code reviews. Maybe shortening release cycles by streamlining development processes is your top priority. Keep in mind that good goals are SMART—specific, measurable, achievable, relevant, and time-bound.

Find the Right Use Case

Determining the right use cases, especially at the outset, is often the most challenging decision when adopting AI in application development. Of course, the use case should align with your goals, but these initial use cases should have a high potential for success and deliver a clear and measurable benefit to the team involved and the organization at large. The key is to avoid trying to automate everything at once. Based on Enterprise Strategy Group research, areas that are highly successful and deliver a clear benefit include:

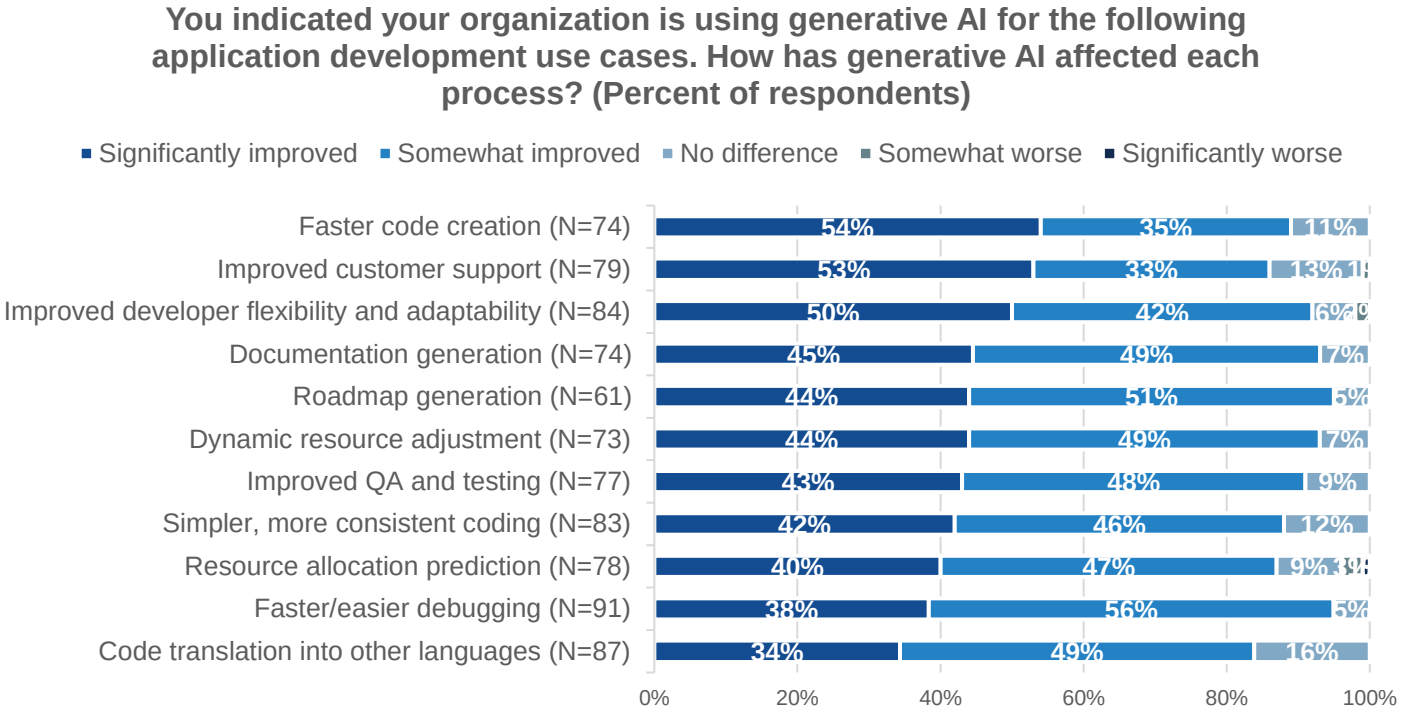
- **Code generation.** As shown in Figure 5, 89% of those who implemented AI in code generation reported demonstrable improvements in coding speed.[7] AI-powered code generation tools can accelerate development by automating repetitive tasks and suggesting relevant code snippets based on context. Leveraging machine learning, AI can analyze existing code patterns and generate code, significantly reducing development time. The high satisfaction with outcomes suggests that this use case delivers tangible results quickly. In addition, AI can be applied here selectively, enabling leaders to compare the improvements of AI-enabled code generation with traditional methods.
- **Documentation generation.** Documenting software code is a time-consuming process and is often viewed as burdensome by software engineers. Generative AI can analyze existing code and comments and create

[7] Ibid.

human-readable text that explains how the code works. It can also tailor that explanation using the vernacular of a software developer. The net effect is increased time available for higher-value work. The standardization can make later debugging easier and the delivery of code changes faster because documentation is generated automatically, saving time. Figure 5 shows that 94% of those who have adopted AI-driven documentation generation for their application development efforts view the results favorably, with 45% saying it "significantly improved" outcomes and an additional 49% saying it "somewhat improved" outcomes.[8]

- **Code translation into other languages.** Maintaining legacy code written in outdated languages can be challenging, and manual translation from one language to another can be time-consuming and error-prone. AI can translate legacy code into a more modern language, making it easier to maintain and update. AI can also potentially help with the performance optimization of the newly coded application, among other things.
- **Test case generation.** By understanding the application's logic and functionalities, AI can generate a wider range of test cases than a human engineer might consider. This can help identify hidden bugs or issues that might have been missed with manual testing alone and ultimately improve software quality.

Figure 5. Successful Outcomes of Implementing AI in Application Development



Source: Enterprise Strategy Group, a division of TechTarget, Inc.

There are many, many more use cases for AI in software development. Selecting the proper use cases is critical to the success of your generative AI mission.

[8] Ibid.

Select the Right Tools and Technologies

Engineering leaders need a strategic approach to selecting the right tools and technologies. This requires a three-pronged strategy: thoroughly researching available options, delivering a proof of concept to validate their suitability, and, finally, closely monitoring and analyzing performance.

Research Available Options

While exploring the diverse and ever-expanding landscape of AI tools and platforms can be overwhelming, considering costs, ease of integration with the development workflow, and scalability will help to narrow down the options and identify the most suitable solution for your specific needs. This can be daunting for engineering leaders who need a deep reservoir of experience in selecting and implementing AI-enabled technologies.

Deliver a Proof of Concept

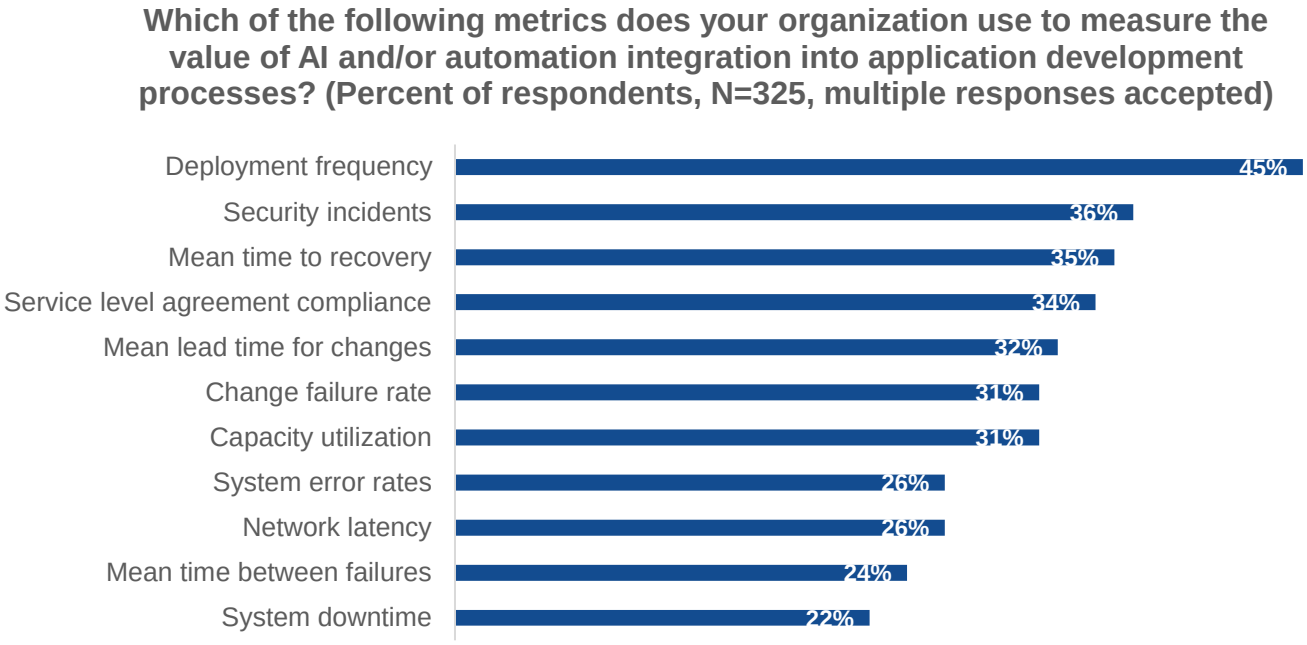
Before committing to full-scale deployment, a proof of concept enables you to test the chosen AI tool(s) with a representative sample of your team's work. This lower-risk approach helps to evaluate the tool's effectiveness, identify any potential integration challenges, and assess its impact on your development workflow.

Once the proof of concept is successfully completed, you're ready to implement and put the AI technology into production.

Monitor and Analyze Performance

Like the adoption of any new process and technology, monitoring and oversight are required to ensure that the results of adoption meet and exceed your expectations for improvement. To this end, a best practice is to monitor and analyze the performance of the AI tool continuously. Track key metrics directly tied to your defined goals, such as those listed in Figure 6 (e.g., code deployment frequency, reduction in security incidents) or time saved for increased developer productivity. Regularly analyze these results to identify areas for improvement within the tool itself or your integration processes. This ongoing assessment ensures that the AI tool remains effective and delivers your envisioned value.

Figure 6. Productivity Improvements and Incident Reduction Lead Value Measurements for AI



Source: Enterprise Strategy Group, a division of TechTarget, Inc.

Refine, Iterate, and Expand

To get maximum value from investment in AI, an interactive process involving software developers, test/QA and other stakeholders is a best practice. By monitoring user feedback, software engineering leaders can increase the value and scope of impact for AI.

Feedback and Improvement

Gather feedback from developers on their experience with the AI tool. This can involve surveys, focus groups, or informal discussions. Analyze their input to identify areas for improvement, such as the tool's user interface, integration with existing workflows, or the types of functionalities offered. Use this feedback to explore modifications within your own development environment to ensure the tool remains aligned with developer needs and delivers the maximum benefit.

Scale Up and Out

As you gain experience and confidence with the initial AI implementation, explore opportunities to expand its use cases. This could involve tackling additional development challenges that align with your overall goals, such as automating more complex tasks, improving code security through AI-powered analysis or incorporating AI-driven test optimization across different parts of your development process. By strategically scaling your AI use cases, you can unlock its full potential to streamline your development workflow and achieve a significant competitive advantage.

Consider Leveraging a Partner for Expertise

It should be relatively clear that the adoption of AI into the application development lifecycle is far from a trivial task. Done incorrectly, an adopting firm can expose itself to undue risk. One approach to mitigating the project risks with AI in application development is to leverage experienced partners to help organizations navigate the complexity of AI. In particular, the benefits of working with a partner can include:

- **Shortened time to value.** Partnering with a technology vendor or servicer with relevant domain knowledge in AI can significantly accelerate your AI adoption and implementation process. These experts can assess your specific development needs and challenges, recommend the most suitable AI tools and technologies tailored to your use case, and provide ongoing support throughout the integration process. This not only saves valuable time and resources in researching and selecting tools but also ensures a smooth and successful implementation, enabling organizations to reap the benefits of AI faster.
- **Minimized risk and maximized ROI.** The landscape of AI tools for application development is evolving rapidly, making it challenging to stay ahead of the curve and select the best solution to meet your needs. Partnering with an experienced or specialized expert can mitigate this risk. They can leverage their expertise to navigate the latest advancements, identify the tools that best align with your specific goals and development environment, and ensure a successful implementation during proof of concept as well as tool rollout. This not only minimizes the risk of choosing an unsuitable tool but also potentially maximizes the return on investment by ensuring the chosen AI solution delivers tangible benefits as quickly as possible for your software development process and team.
- **Knowledge transfer and skills development.** As highlighted in Figure 2, above, a significant challenge for organizations implementing AI in application development is the lack of in-house skills and expertise. This lack of skills can hinder the existing team's ability to utilize AI tools and technologies effectively. Partnering with an AI specialist can bridge this skills gap by providing the necessary knowledge and experience to navigate AI adoption. Many partner organizations also offer end-user and administrator training to ensure that the skills needed to utilize and maintain the technology become part of the organization's native capabilities, ensuring that skills gap issues and their potential impact on productivity, security, and workflows are minimized.

AWS Partner Spotlight: CircleCI

CI/CD and Generative AI

Continuous integration/continuous delivery (CI/CD) is a vital element in the software development lifecycle for generative AI applications because it accelerates time to market, enhances code reliability through automated testing, and improves AI model accuracy and security. As a long-established leader in CI/CD, CircleCI is well-positioned to team with AWS to deliver substantial benefits for any organization developing and deploying generative AI applications.

CircleCI and AWS

Since its establishment more than a decade ago, CircleCI has worked closely with AWS across a wide range of solutions, including SageMaker, Bedrock, and now Amazon Q. CircleCI is a recognized top AWS technology partner and holds the AWS DevOps Competency. Its long-time relationship with AWS is important in its CI/CD and application testing work in AWS environments because the company has built an extensive array of knowledge of how applications perform—and how they should perform—in AWS environments.

CircleCI's work with AWS in automated testing and validation is essential in evaluating and verifying appropriate generative AI model performance. CircleCI enables the development of test cases to determine whether various content forms such as text and images are "safe" and to achieve optimal quality and reliability. CircleCI can also be integrated with other AWS services to ensure proper cybersecurity, governance, and regulatory compliance when dealing with sensitive and personally identifiable information.

Solving Customers' AI/ML Pain Points

This showcase highlights the generative AI use cases developers need and the current DevOps issues teams want to solve. CircleCI addresses many of these pain points and provides AWS-integrated solutions for the top 5 generative AI use cases identified in Figure 1, including improved QA and testing, faster/easier debugging, improved developer flexibility, faster code creation, and simpler, more consistent coding.

CircleCI is a leader at building and testing organizations' software, whether a human or AI writes it. And, as more software is designed as an agentic AI system, CircleCI is the best tool to test those models, as well as the applications.

In addition, CircleCI delivers significant efficiencies for 8 of the 12 most time-consuming processes detailed in Figure 4, including testing and quality assurance, integration, CI/CD pipeline configuration, security, regulatory compliance, performance optimization, debugging and issue resolution, and release and error resolution.

As a software-as-a-service offering, CircleCI is naturally suited for AWS environments because of its cloud-centric focus that helps organizations get started quickly and easily on testing and validating generative AI application code and models. CircleCI does this in a way that is secure, fast, simple to integrate, and able to reduce administrative complexity and overhead.

Conclusion: Embracing AI in Software Development With Confidence

The software development landscape is evolving rapidly. While AI offers tremendous potential for accelerating and improving SDLCs, organizational leaders acknowledge the associated risks. This paper has explored the benefits of AI in development, from intelligent code assistants to streamlined testing processes and more. However, navigating the path toward safe and effective AI adoption requires a strategic approach. Partnering with experienced AI specialists like Amazon Web Services and its approved partners can bridge critical skill gaps and provide the expertise needed to navigate AI adoption. By leveraging this expertise, organizations can mitigate risks associated with AI implementation, maximize the benefits of AI, and prepare for a future powered by AI, ensuring they are well positioned to leverage its full potential for long-term success.

By taking a proactive approach to AI adoption with the right guidance and support, organizations can unlock the competitive advantage AI offers in today's competitive software development landscape, paving the way for a future where software engineers are freed to produce innovative, secure, maintainable applications of high quality that increasingly serve as the linchpin of competition and business value creation.

About Enterprise Strategy Group
TechTarget's Enterprise Strategy Group provides focused and actionable market intelligence, demand-side research, analyst advisory services, GTM strategy guidance, solution validations, and custom content supporting enterprise technology buying and selling.

contact@esg-global.com
www.esg-global.com