



Modern DevEx and Classic Tech: Unlocking Developer Experience for IT Teams

A practical guide to transforming DevEx for teams working with IBM Z[®], IBM[®] i, and other established technologies



Contents

- 04 What is DevEx?
- 05 Core components of DevEx
- 05 Expanding the scope of DevEx
- 06 Value of DevEx
- 07 The evolution of DevEx
- 09 AI in development: A tool, not a competitor
- 10 Application insights
- 11 Design, define, collaborate
- 12 Developer assistance
- 15 DevEx for organizations running established technology



Introduction

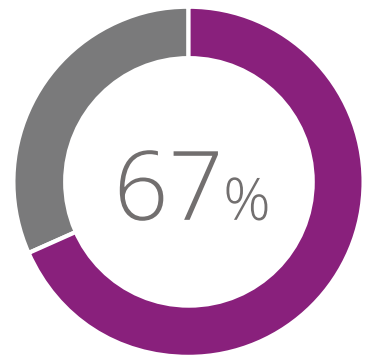
The role of a software developer has never been more dynamic or demanding. Technology evolves at an unprecedented pace, as do the challenges and expectations placed on developers — from the rapid adoption of AI-driven development tools to the increasing complexity of modern software stacks. Simultaneously, the realities of tech sector layoffs, high-pressure work environments, and the relentless need to upskill all create an environment where burnout is far too common. According to a recent report, 67% of developers have left a job due to the pressures of deployment errors or know someone who has, while 61% cite cumbersome internal processes as barriers to innovation ([WorkLife](#)).

Software development may actually be evolving too rapidly. We now ship applications that are often 90% dead weight, prioritizing the speed of feature delivery over efficiency and optimization. Over the last two decades, development has become faster, more accessible, and more automated — allowing businesses to produce applications with more features than ever, often built by increasingly less experienced developers ([Stack Overflow](#)). Meanwhile, emerging technologies like WebAssembly, ESBuild, SWC, Bun, and Yew are pushing the boundaries of performance, and lightweight frameworks like Tauri are providing alternatives to bloated software packaging. These advances signal a shift in how software is built. But they also present new challenges in maintaining code quality, managing complexity, and keeping developers engaged and productive.

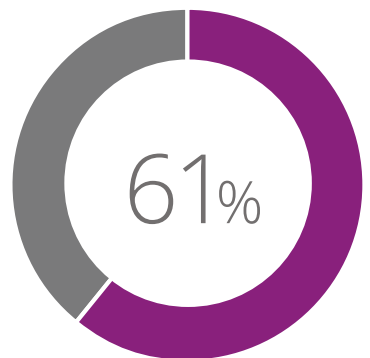
Despite these challenges, demand for developers continues to rise. According to the U.S. Bureau of Labor Statistics, employment for software developers, QA analysts, and testers is projected to grow 25% from 2022 to 2032, adding roughly 153,900 new job openings annually ([Jessup University](#)). This demand is fueled by the expanding role of AI, automation, cybersecurity, and mobile applications in nearly every industry. However, meeting this demand isn't as simple as hiring more developers. Different generations — from baby boomers to Gen Z — approach work differently, which creates varying expectations for tools, processes, and work-life balance. Likewise, some newer developers prioritize rapid learning and collaboration, while some experienced developers put a premium on stability and efficiency. These differences influence the developer experience in profound ways, shaping the tools, workflows, and organizational cultures that drive modern software development.

Enter DevEx.

A strong Developer Experience (DevEx) strategy is essential for companies seeking to retain talent, drive innovation, and maintain software quality. This whitepaper explores the evolving landscape of Developer Experience, the role of AI and automation, the key factors shaping the future of software development, and how organizations running critical enterprise systems and technology can create an environment where developers focus on high-impact work, stay engaged, and build the future of technology.



Developers who have left a job due to the pressures of deployment errors or know someone who has.



Cite cumbersome internal processes as barriers to innovation.

What is DevEx?

Developer Experience (DevEx) is a comprehensive discipline that encompasses the tools, processes, workflows, and culture that define a developer's day-to-day work. A strong DevEx strategy ensures that developers can focus on solving problems and building high-quality software without being hindered by inefficiencies, outdated systems, or bureaucratic hurdles. When organizations invest in DevEx, they create an environment where developers are more productive, engaged, and innovative, leading to faster software delivery, improved product quality, and greater business success.

Beyond just optimizing individual workflows, DevEx also fosters a thriving developer community by promoting seamless collaboration, reducing cognitive load, and enabling a culture where developers feel empowered to experiment, iterate, and continuously improve. By prioritizing developer well-being, efficient tooling, and streamlined processes, organizations can create a virtuous cycle — where satisfied developers produce better software, which in turn drives business growth and customer satisfaction.



Core components of DevEx

01

Tools and infrastructure

A developer's efficiency is directly tied to the tools and infrastructure they use daily. Well-integrated and reliable tools minimize friction in development, allowing developers to code, test, and deploy efficiently. Tools include IDEs, SDKs, version control systems, CI/CD pipelines, cloud platforms, containerization technologies, and automation frameworks. However, DevEx is not just about providing tools. It also offers flexibility and choice, allowing developers to adapt workflows to their preferences and expertise. A modern DevEx strategy ensures that tools integrate seamlessly, reduce manual work, and enable a frictionless development environment.

02

Documentation and knowledge resources

Developers rely on clear, well-structured, and up-to-date documentation to navigate APIs, frameworks, internal codebases, and development environments. A strong DevEx ensures that documentation is easily accessible, regularly maintained, and supplemented with interactive examples, tutorials, and AI-assisted search tools. Investing in high-quality documentation reduces onboarding time, improves collaboration, and prevents unnecessary delays caused by knowledge gaps.

03

Onboarding and continuous learning

A smooth onboarding process is critical to helping developers become productive quickly. Well-defined onboarding pathways, structured mentorship programs, interactive training, and hands-on coding exercises allow developers to ramp up effectively. Beyond onboarding, there are continuous learning opportunities. Knowledge-sharing sessions, hackathons, internal workshops, and access to industry certifications all ensure that developers stay ahead of industry trends, upskill efficiently, and maintain long-term engagement.

04

Team and system processes

The way teams prioritize work, conduct code reviews, handle deployments, and manage collaboration has a direct impact on developer experience. Inefficient, bureaucratic processes can slow innovation and frustrate developers, while well-structured workflows enable agile development, faster iteration, and more predictable outcomes. DevEx initiatives focus on removing blockers, optimizing internal systems, and ensuring that development, testing, and deployment processes align with best practices for efficiency, security, and scalability.

Expanding the scope of DevEx

While traditional discussions around DevEx often focus on tooling and onboarding, developer culture and well-being are equally important. Reducing unnecessary meetings, eliminating excessive bureaucracy, and fostering a culture where developers feel comfortable innovating and taking risks are critical to a thriving developer experience. Additionally, feedback loops — such as automated testing insights, real-time error tracking, and peer code reviews — provide developers with meaningful, actionable insights into their work.



Value of DevEx

Enable DevOps

One of the key advantages of a strong DevEx strategy is its ability to enable DevOps. With DevOps as a standard practice, developers build, test, and deploy applications in consistent, reproducible environments, significantly reducing the challenges associated with software delivery. This approach enhances collaboration between development and operations teams, simplifies deployment troubleshooting, and minimizes downtime. As modern DevOps practices evolve, the traditional distinctions between developers, testers, and operations personnel are becoming increasingly blurred. A well-implemented DevEx strategy facilitates this transition by fostering cross-functional collaboration, breaking down silos, and ensuring that teams can work together efficiently to deliver high-quality software.

Developer productivity and software quality

Another core benefit of DevEx is its impact on developer productivity and software quality. When developers have access to well-integrated, user-friendly tools and optimized pipelines, they focus more on building and refining applications rather than dealing with cumbersome processes. Automated pipelines streamline software delivery by handling testing, deployment, and monitoring, which reduces manual errors and accelerating time to market. As a result, developers produce higher-quality, more secure applications that contribute to better customer satisfaction and improved business outcomes.

Modernization without disruption.

A DevEx strategy is essential for modernizing without disruption. Many organizations face the challenge of updating their foundational technology without completely rewriting applications — a costly and time-intensive process. By addressing the skills gap through improved tooling, automation, and learning resources, companies integrate modern development practices with existing infrastructure. This allows businesses to modernize incrementally, ensuring continuity while progressively enhancing their systems.

Deployment team efficiency

A strong DevEx strategy improves deployment team efficiency by reducing bottlenecks and ensuring smoother handoffs between development and operations. This means fewer deployment-related incidents, less time spent troubleshooting, and an overall faster, more reliable release process.

Retain top talent

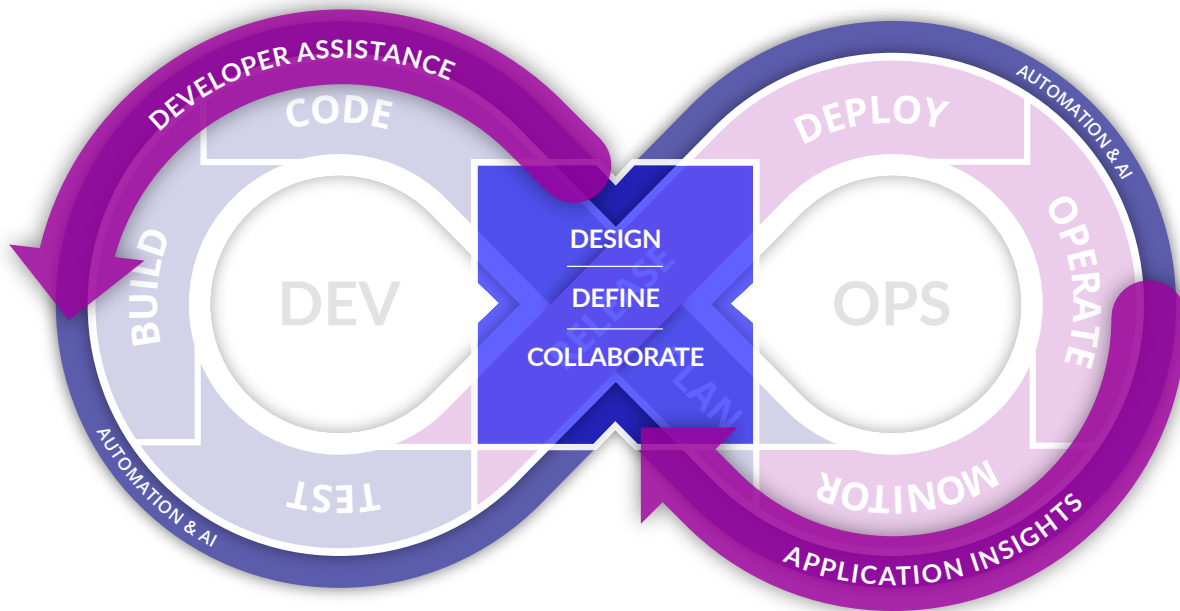
Organizations that prioritize DevEx tend to retain top talent more effectively. Developers are more likely to stay with companies that provide modern tools, streamlined workflows, and a culture that values their contributions. This reduces turnover and the associated costs of hiring and training new engineers.

Fuel innovation

A well-executed DevEx strategy fuels innovation by removing friction from the development process. When developers spend less time on tedious tasks and operational roadblocks, they have more bandwidth to experiment, iterate, and develop creative solutions. This leads to a more dynamic, forward-thinking development culture that drives business growth and competitive advantage.

The evolution of DevEx

The future of developer experience (DevEx) is poised to be transformed by advancements in automation, AI, and collaborative tools, all which emphasize efficiency, accessibility, and innovation. Here's what it could look like:



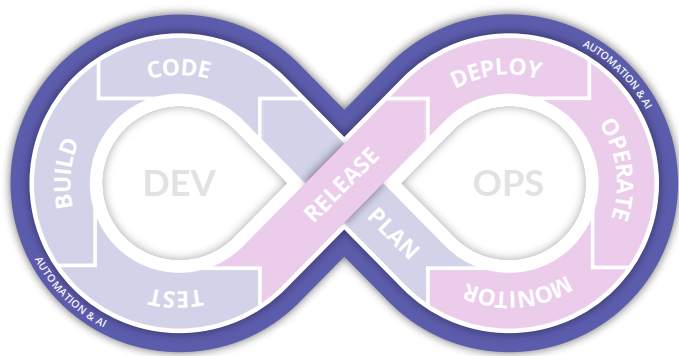
The DevOps process itself will continue as it has been: **Plan. Code. Build. Test. Deploy. Operate/Monitor** — and so on, as needed. What changes is how developers experience this process. As visibility and insights increase, the boundaries between functions throughout the DevOps process soften when it comes to human work to be done. Automation and AI speeds up feedback, provides more insightful guidance, and empowers closer collaboration across teams. This evolution streamlines the whole DevOps process, speeding up time to market and time to innovation while improving team productivity.

We use the term DevEx (developer experience) but this evolution will impact developers and every worker engaged in the DevOps process, directly or indirectly. The result will be an expanded team that has become symbiotic, operating with organic ebbs and flows of resources that target specific needs and ensure that they are met on time.

DevEx is data-driven insights

The future of Developer Experience (DevEx) will be driven by data-driven insights. Advanced development analytics will provide real-time visibility into bottlenecks, helping teams proactively address slowdowns in coding, testing, and deployment processes. By tracking key metrics — such as time spent on debugging, build failures, and deployment frequency — organizations optimize development cycles, reduce friction, and enhance overall productivity. These insights will empower engineering leaders to make informed decisions and ensure that teams work smarter, not harder, while maintaining high standards of code quality and performance.

Beyond productivity, data-driven tools will enhance collaboration and code quality, ensuring that teams operate cohesively and efficiently. Metrics tracking pull request activity, review times, and knowledge sharing will highlight opportunities to improve developer collaboration and onboarding processes. AI-driven analytics will also help identify patterns in code quality, flagging potential vulnerabilities or inefficiencies before they become major issues. As DevEx evolves, this continuous feedback loop of insights and optimizations will transform development into a more transparent, adaptive, and intelligent process. Teams will build better software, faster and with fewer roadblocks along the way.



Automation

Woven throughout the DevOps loops is automation and AI.

Seamless automation

Automation, in its traditional form or as genAI, underlies nearly every component in the future of DevEx. The new ability to automate certain aspects of the DevOps process changes the speed at which code reaches production. Teams become more productive. Insights that would've taken weeks to discover manually are uncovered in seconds.

Seamless automation eliminates friction in software development workflows. As development environments become increasingly complex — spanning cloud, on-premises, and hybrid ecosystems — automation ensures smooth transitions between coding, testing, deployment, and monitoring.

Automation and AI will change what it means to be innovative and disrupt market competition entirely.

By integrating seamless automation into CI/CD pipelines, infrastructure management, and security compliance, organizations empower developers to work more efficiently, reducing cognitive load and context switching. The result is a streamlined experience where development cycles are faster, more reliable, and free from bottlenecks that traditionally slow down productivity.

AI in development: A tool, not a competitor

The rise of AI in software development has sparked both excitement and concern. History shows that every major technological advancement — from industrial automation to cloud computing — has been met with skepticism. Yet, the [automation paradox](#) tells us that rather than replacing human expertise, technology often increases the demand for skilled professionals who can use it effectively. AI should be viewed as a powerful tool that enhances, rather than replaces, human developers, enabling better efficiency across the entire DevOps spectrum.

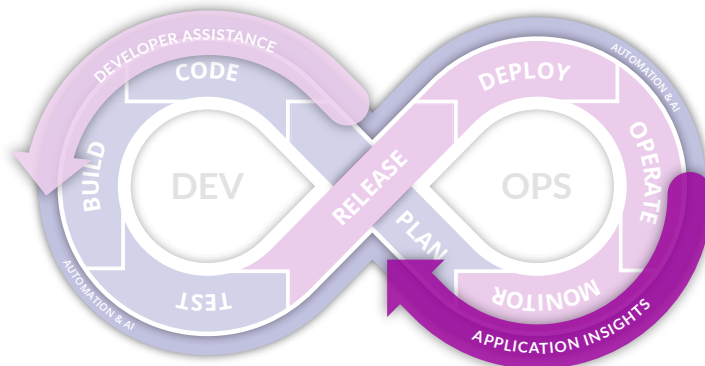
Unlike Artificial General Intelligence (AGI), which would hypothetically think and reason like a human, today's AI remains limited to specific tasks. It excels at pattern recognition, automation, and augmentation, but struggles with complex problem-solving, strategic thinking, and nuanced decision-making. Developers must still design architecture, optimize performance, and ensure maintainability — tasks that AI cannot fully automate.

The key to success lies not in blindly relying on AI but in strategically leveraging it to improve workflows, maintain high code quality, and drive innovation. Organizations that embrace AI thoughtfully — understanding its strengths, limitations, and best applications — will position themselves for a more efficient and scalable future.

The key to success lies not in blindly relying on AI but in strategically leveraging it to improve workflows, maintain high code quality, and drive innovation.

Application insights

Modernization teams often find themselves in a state where they recognize the need to modernize but not the way to start. This is especially daunting when you're managing monolith applications developed over decades.



With application insights, modernization teams can answer questions like:

What are the most commonly used functions in this system?

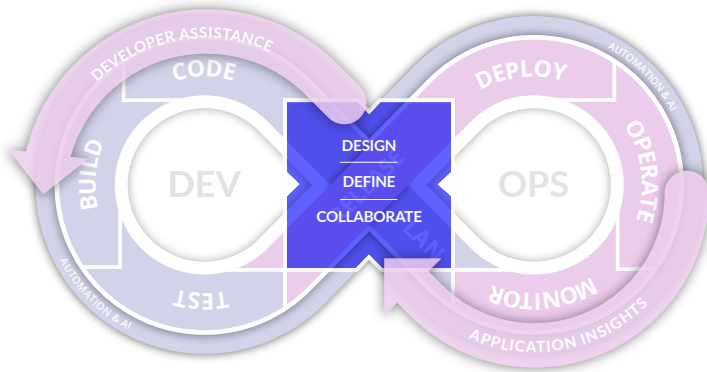
How does this application architecture fit into the broader ecosystem?

How do internal libraries and customized coding standards impact development?

Application insights can also provide automated deployment risk analysis or dependency mapping and impact analysis, enhancing observations from the Ops side of the DevOps process: deploy; operate; and monitor.

But this level of intelligence goes beyond syntax and code snippet. It enables developers to work effectively within complex, customized enterprise systems. The insights made available are not just from static analysis of the application

but also includes dynamic insights into how the business is truly using the application, above and beyond its original intent. This process-discovery approach to application insights provides organizations with a clear roadmap for modernization. The outcome is a focusing of resources on modernization projects that create business value, drive innovation, and enhance the customer experience — all without unnecessary disruption.

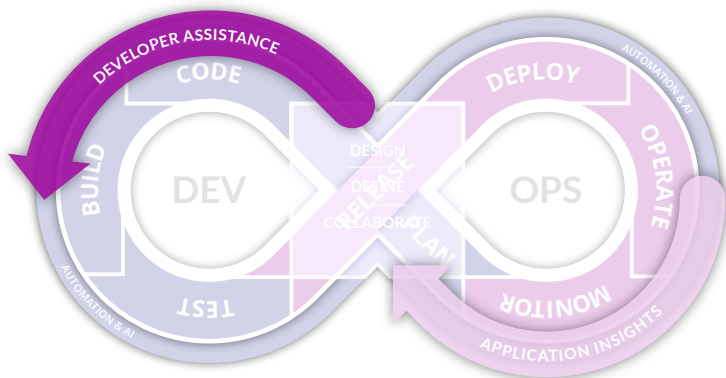


Design, define, collaborate

Enhanced collaboration: There will be an acceleration of the standard feedback we see today. The operations team, for example, will be able to notify development about a configuration that needs fixing before deployment happens. We'll also see customers and support teams leveraging AI tools to better communicate to product what needs enhancements. As a result, product management could mockup a prototype or a UI quickly and send to the development or UX team, respectively, to kickstart a project. The ability to communicate with higher quality information to another DevOps team member is core to the future of DevEx, upgrading how teams work together to scale and deploy innovative code.

Enhanced collaboration enables seamless teamwork across distributed teams. Virtual development spaces, such as GitHub Codespaces, eliminate the traditional setup overhead by providing cloud-based, pre-configured environments for developers to instantly access and share. This ensures consistency across projects and allows teams to collaborate in real time, regardless of location. By reducing dependency on local machine configurations and streamlining onboarding, developers can focus on building solutions instead of troubleshooting environment inconsistencies. As remote and hybrid work models continue to rise, virtual development spaces will become essential for maintaining productivity and cohesion in globally distributed teams.

The next evolution of real-time co-development will further enhance DevEx by bringing the Google Docs-style collaboration experience to coding. Emerging platforms will allow multiple developers to write, review, and debug code simultaneously, offering immediate feedback and reducing iteration cycles. This shift will improve code quality and accelerate innovation by fostering continuous knowledge sharing and peer-driven learning. As pair programming tools become more prevalent, real-time collaboration will evolve into a dynamic, interactive process where developers can solve complex problems together with minimal friction. In this future, coding will no longer be a solitary activity but a connected and interactive experience that enhances both efficiency and creativity.



Developer assistance

The goal of developer assistance tools is to focus the developer's time on coding, removing tedious work and increasing both code productivity and the quality of the coders' work experience while reducing the risk of churn.

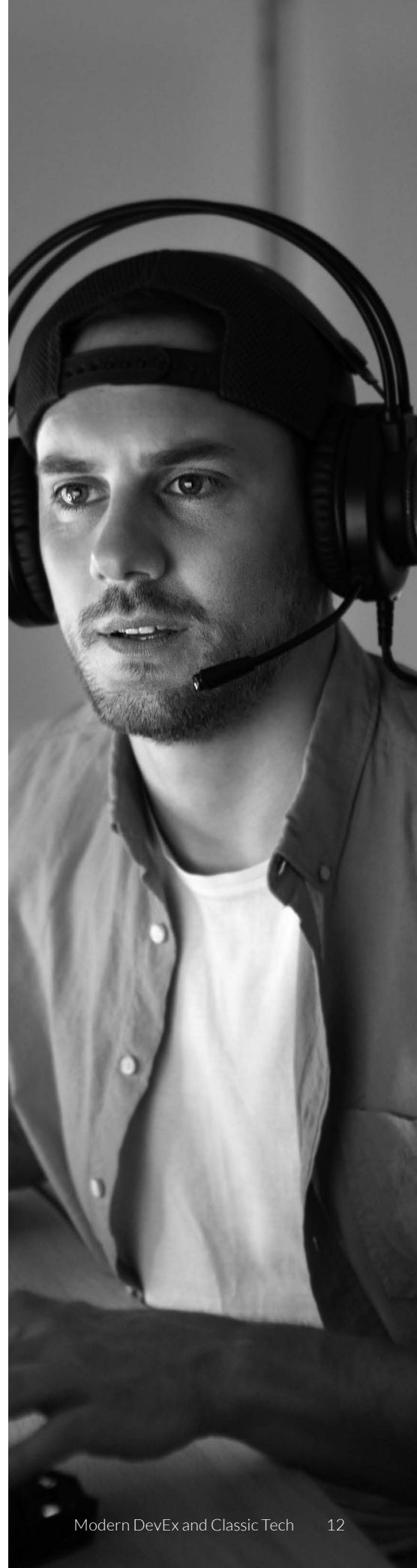
Developer assistance can take ownership of tasks like comment writing and test script creation. Developer assistance can help less experienced developers understand the code and more experienced developers remember what the code does. Assistants can also quickly find code that can be replicated and make recommendations on what code to build. Developers will receive more high-quality insights from different teams across the DevOps process, including operations and support.

In addition, today's developers have several tools they work in. Some work in Integrated Development Environments (IDE) like VS Code, going to Jira to manage bugs, Jenkins to deploy the code, and Git to manage code versions. Some of those tools do have integrations. But integrations are not a 1:1 replacement for a single holistic experience.

AI-powered development assistance

AI will seamlessly integrate into modern software development pipelines. Beyond code assistance, AI-powered insights can identify duplicated code, detect inefficiencies, and suggest improvements before issues escalate. AI, when combined with existing observability and application performance monitoring tools, can help development teams refine their processes and deliver more reliable software. Some examples:

- **Code generation and debugging:** AI will further integrate into IDEs, providing real-time code suggestions, auto-completion, debugging, and error resolution. Tools like GitHub Copilot and OpenAI Codex are early examples.
- **Test generation:** AI will play an evolving role in test generation, starting with test automation through to intelligent code fix where the AI model understands the code and the intent of the code well enough to fix the code itself.



- **Natural language interfaces:** Developers may interact with codebases through conversational AI, asking for code snippets or explanations in plain language.
- **UX requirements:** AI can help product and UX teams identify requirements for an user experience, including automated mock-ups of the UI.
- **Prototypes:** Product management could quickly build prototypes they hand off to developers to better communicate the intent and goals of the development work.

Integrated learning and upskilling

Integrated learning and upskilling ensure that developers have seamless access to knowledge exactly when they need it. Modern development platforms are embedding real-time documentation, tutorials, and learning modules directly into workflows. This reduces the need for developers to switch contexts or search for external resources. Inline code suggestions, interactive API documentation, and AI assistants providing relevant examples are all tools that help developers quickly overcome roadblocks and stay productive. This shift from passive to embedded learning enhances efficiency. It also allows developers to continuously grow their skills while coding rather than relying solely on external training sessions.

As DevEx evolves, adaptive training models will take developer upskilling to the next level:

- Learning experiences will be tailored to individual performance and interests.
- AI-driven platforms will analyze coding patterns, feedback loops, and skill gaps to provide personalized recommendations for new frameworks, best practices, and emerging technologies.
- Developers will receive curated learning paths that align with both their career goals and business needs, ensuring continuous growth without disrupting their daily work.

This dynamic, on-the-job learning approach will not only improve developer proficiency but also future-proof teams by enabling them to quickly adapt to new tools, methodologies, and industry trends.

...instead of replacing traditional coding, these platforms will complement it — allowing developers to seamlessly dive into custom code for fine-tuning, optimization, and deeper integrations.

Low-code/No-code synergy

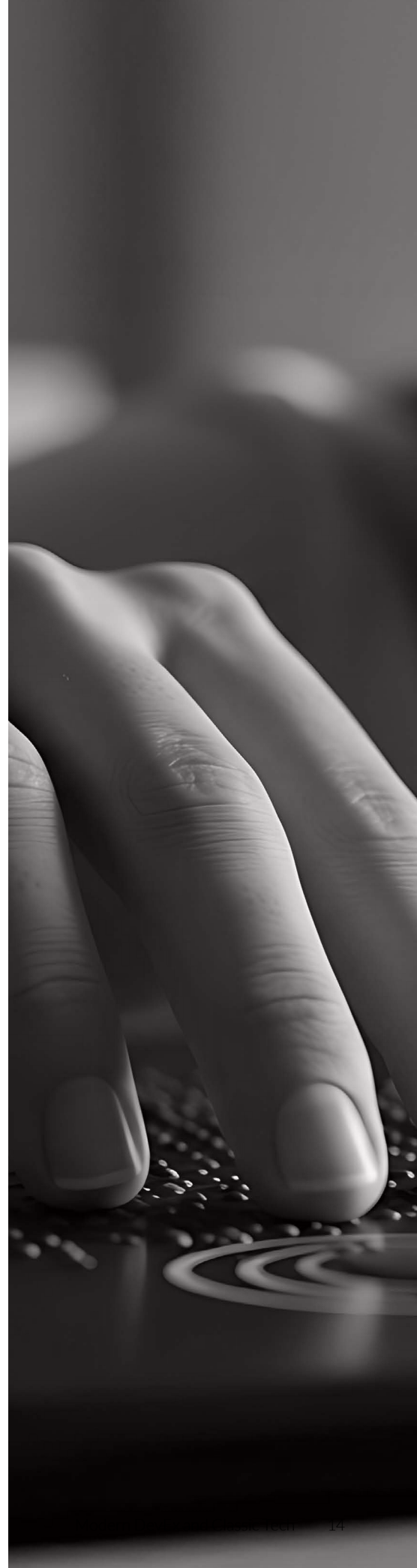
The synergy between low-code/no-code and traditional development enables hybrid workflows that balance speed and flexibility. Developers will increasingly leverage low-code tools to accelerate prototyping, quickly building functional models without starting from scratch. However, instead of replacing traditional coding, these platforms will complement it — allowing developers to seamlessly dive into custom code for fine-tuning, optimization, and deeper integrations. This hybrid approach streamlines development cycles, reduces time-to-market, and ensures that teams can focus on delivering high-impact solutions without being slowed down by repetitive or boilerplate coding tasks.

As low-code and AI-assisted development become more sophisticated, software creation will become more democratized. A broader range of contributors participating in the development process is the result. Developers will shift their focus from routine maintenance and manual coding to more strategic problem-solving, such as architecting scalable solutions and enhancing system performance. Meanwhile, the convergence of automation, observability, and intelligent workflows will further enhance DevEx so developers can work within an intuitive ecosystem that proactively optimizes their experience. This evolution will turn software development into a fluid and highly efficient process: Automation handles repetitive tasks. Low-code simplifies complexity. Developers concentrate on innovation.

Inclusive and accessible development

The future of Developer Experience (DevEx) will be shaped by inclusive and accessible development. All developers, regardless of ability or background, will be able to fully participate in software creation. Development tools and platforms will integrate voice control, screen readers, and improved UX for visually impaired developers, making coding more accessible than ever before. Other features will include adjustable contrast modes, keyboard navigation enhancements, and AI-powered code assistants. As these accessibility innovations become standard, organizations will unlock a more diverse talent pool and foster innovation through broader perspectives and inclusive design practices.

Beyond accessibility, localization and language adaptation will play a key role in making development tools more globally inclusive. Platforms will offer real-time translations, multilingual coding documentation, and cultural adaptations that empower developers from different regions to work more efficiently. By reducing language barriers and creating accessibility for all, the next generation of DevEx will position developers of all abilities and backgrounds to contribute at the highest level and drive more collaborative, globally connected development ecosystems.



DevEx for organizations running established technology

All of this sounds great theoretically, but much of this can feel out of reach for organizations running foundational systems such as IBM Z®, IBM® i, MultiValue, and Uniface. Below we review six steps that organizations can take today to take advantage of these trends and deliver value to the business even when running on established platforms.

Continue To Move Toward DevOps Best Practices

For established platforms — especially organizations running mainly on-premises systems and software, development and operations still run separately to some capacity. Over the last several years, heritage tech teams have experienced increasing pressure to perform like other DevOps teams. That means:

- Agile development cycles with fast response times to market and customer needs, and quality code in a multi-code environment
- Close collaboration between development and operations/IT for better visibility across the entire development and deployment process, tighter controls of sensitive code and data
- Coordinating with another team — for example, Java developers — across the DevOps process
- New stakeholders. Applications running on or through established platforms no longer live in silos and are being pulled into new workflows and UI/UX
- Recruiting younger developers who have a new, modern view of what development looks like, what tools they need, what methodology works, etc.

The more you can evolve to a coordinated DevOps approach, the easier it will be to take advantage of the new technology and trends like DevEx, AI, etc.

Approach Modernization as a Mindset

Modernization isn't just about upgrading enterprise systems or adopting the latest technologies. It's a mindset that embraces continuous improvement, agility, and innovation. Organizations that succeed in modernization don't view it as a one-time project but as an ongoing process that requires adaptability and a culture of learning. This mindset encourages leaders and teams to challenge outdated practices, embrace emerging technologies, and refine workflows to meet evolving business needs. Without this cultural shift, even the most advanced tools and platforms can become underutilized, misaligned with business goals, or resisted by the workforce.



Move Out of the Silo

Organizations rely on a growing number of software applications, cloud services, and data platforms to drive their business operations. But when these systems operate in isolated silos, inefficiencies emerge: Data fragmentation. Manual workarounds. Security vulnerabilities. And poor cross-functional collaboration. To unlock true operational efficiency, businesses must integrate their IT ecosystems so critical applications work together seamlessly.

Without proper integration, essential business processes such as HR, finance, customer relationship management, and IT service management remain disconnected. Key enterprise tools like Workday, Salesforce, and ServiceNow hold vital business data. But without synchronization with core applications, organizations risk duplication, misalignment, and slower decision-making. By integrating IT systems, businesses running on established platforms can:

- Enable seamless data flow across departments, reducing redundancy and improving accuracy.
- Minimize manual intervention and human error.
- Enhance customer experience for faster, more personalized service delivery.

Integration is essential. But businesses must also ensure that what makes their applications unique — like proprietary business logic — remains intact. Modernization is not about discarding the past. It's about connecting the old with the new in ways that drive business efficiency, agility, and long-term growth. Organizations that embrace this mindset will be well-positioned to navigate the challenges of an ever-evolving digital landscape.

Standardize

In many organizations, critical knowledge about systems and applications exists only in the minds of developers, some of whom have been writing code the same way for decades. Their expertise is invaluable. But relying on tribal knowledge and outdated processes creates risks: Key insights become siloed. Onboarding new developers is difficult. And modernization efforts stall. To build a more resilient and scalable development environment, organizations must embrace standardization — not as an inflexible rulebook, but as a framework for efficiency, collaboration, and continuous improvement.

A shift toward modern development best practices is essential in this transformation. Many companies still operate on long, multi-year release cycles, which delay innovation and create unnecessary complexity. Not every organization needs to adopt the rapid iteration of tech startups. But moving from deploying updates every two years to a more manageable six-month cycle can significantly improve responsiveness and competitiveness. This approach ensures that systems evolve alongside business needs, reducing the burden of massive, disruptive overhauls.



In addition, standardizing developer tools like Git, Jenkins, and other components of the software delivery toolchain is a foundational step toward improving DevEx. When teams use a consistent set of tools, it reduces friction in onboarding, enables smoother collaboration, and eliminates redundant troubleshooting across environments. Developers spend less time navigating tool differences and more time building and delivering software. This consistency also opens the door to automation, streamlined workflows, and shared best practices — paving the way for platform engineering, internal developer portals, and other DevEx-enabling strategies. Ultimately, tool standardization empowers organizations to scale development more efficiently and foster a culture where developers can thrive.

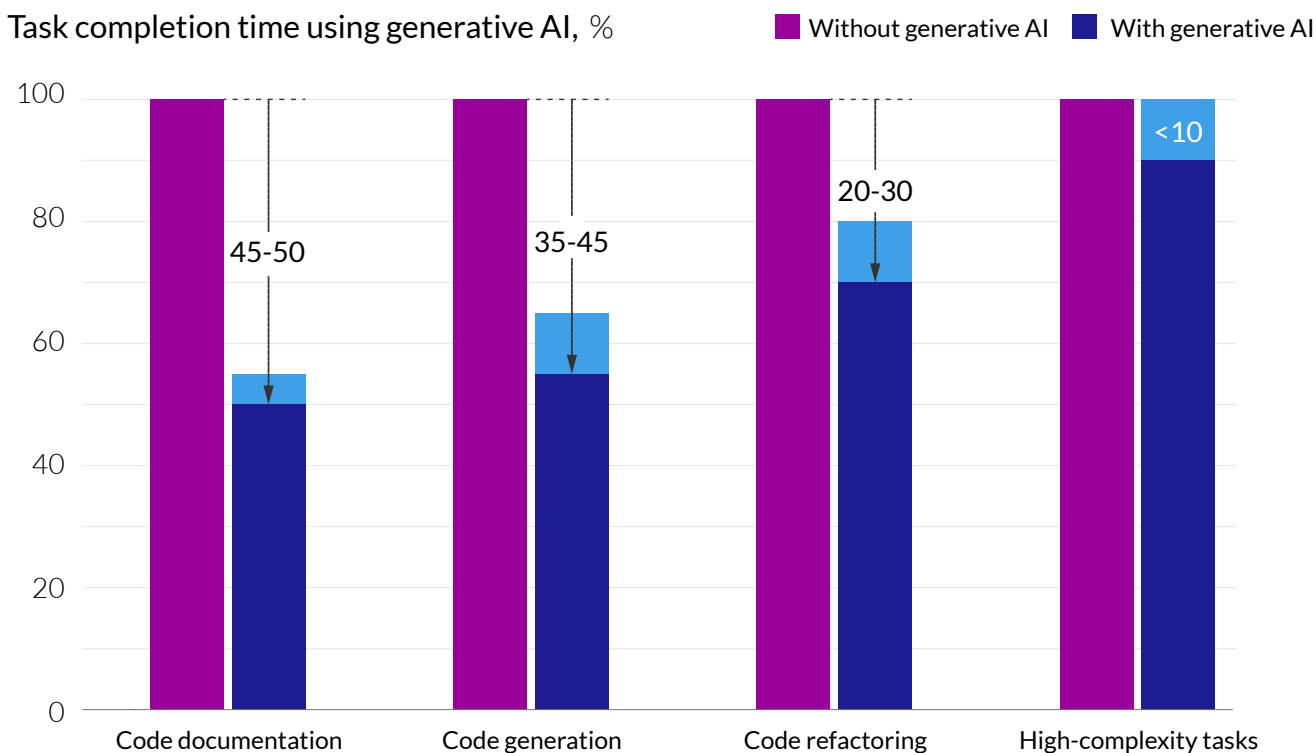
AI-powered development tools further accelerate this transition. But they work best when supported by standardized processes and tools. Without clear coding standards, documentation practices, and automation frameworks, AI can amplify inconsistencies rather than solve them. By establishing uniform development environments, version control strategies, and structured workflows, organizations can leverage AI more effectively. This includes code assistance, process automation, or quality control because standardization isn't about limiting flexibility. It's about creating a foundation that supports innovation, reduces technical debt, and enables developers to focus on delivering value rather than reinventing the wheel.

Cultivate talent

The developer workforce is evolving. And organizations that rely on established platforms must recognize how generational shifts impact talent acquisition, retention, and innovation. Newer developers expect the same modern tools they use in cloud-native environments such as VS Code, Git, and CI/CD pipelines. Organizations won't convince young developers to code in green screen environments. But they can integrate foundational systems with modern tools to make them accessible and relevant. Meanwhile, experienced developers who have spent years working with established platforms hold critical institutional knowledge but may be resistant to change. It may be difficult to convince seasoned developers to work in modern GUIs but they'll still have access to the green screen they know and love (and likely have hundreds of macros saved to make them power developers). Successfully bridging these perspectives requires modernizing both technology and processes. Create a development environment that supports both existing expertise and new talent expectations.



Generative AI can increase developer speed, but less so for complex tasks.



Explore and grow with AI

Learn how to leverage today's AI tools. Play with CoPilot, IBM Watson X Code Assist, openAI, Codium, or one of the many other options at your fingertips. Consider the ways they can help improve your ability to get high quality code out the door quickly. How can your teams take advantage? Where does it make the most sense to plug AI into the workflows?

A [report](#) from McKinsey suggests that developers who aren't using generative-AI tools are missing out and tend to be less happy than the developers who are not missing out.

For technical teams working on heritage technologies while often backlogged with months of maintenance work, DevEx opens doors to more fulfilling projects. This not only creates environments with lower attrition but also frees teams to pursue more creative work, which fosters innovation and competitive strength in the market.

But innovation only comes when organizations and technical teams are willing to adapt to new opportunities. This is the time to pilot tools and process, watch changes in the market, and recalibrate with an agile mindset.

[Learn more about AI and Developer Efficiency >](#)

Conclusion

The future of software development will be defined both by the technologies we adopt and by the experiences we create for the developers who build with them. As organizations navigate a landscape shaped by rapid innovation, AI, automation, and evolving workforce expectations, Developer Experience (DevEx) emerges as a strategic imperative — not a luxury. By focusing on seamless tooling, inclusive processes, intelligent automation, and continuous learning, organizations can unlock the full potential of their teams, regardless of whether they're working in cloud-native stacks or on established platforms like IBM i and IBM Z. With the right mindset and practical steps — from standardizing development environments to integrating modern workflows — DevEx can bridge the gap between established platforms and more modern solutions, reduce friction, and create the kind of developer-first culture that drives long-term innovation, productivity, and resilience.



About Rocket Software

Rocket Software is the global technology leader in modernization and partner of choice that empowers the world's leading businesses on their modernization journeys, spanning core systems to the cloud. Trusted by over 12,500 customers and 750 partners, and with more than 3,000 global employees, Rocket Software enables customers to maximize their data, applications, and infrastructure to deliver critical services that power our modern world. Rocket Software is a privately held U.S. corporation headquartered in the Boston area with centers of excellence strategically located around the world. Rocket Software is a portfolio company of Bain Capital Private Equity. Follow Rocket Software on [LinkedIn](#) and [X](#).



Modernization. Without Disruption.™

Visit RocketSoftware.com >

© Rocket Software, Inc. or its affiliates 2025. All rights reserved. Rocket and the Rocket Software logos are registered trademarks of Rocket Software, Inc. Other product and service names might be trademarks of Rocket Software or its affiliates.

IBM® i and IBM Z® are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide.

MAR-13791_WP_DevExLeadership_V5

