

# The performance engineering playbook for outpacing complexity

AI-powered and hybrid-ready, so confidence  
is built into every release

# Contents

|                                                             |    |
|-------------------------------------------------------------|----|
| Applications don't fail quietly anymore                     | 3  |
| When "it works" isn't the same as "it holds up"             | 4  |
| Where AI does the heavy lifting                             | 5  |
| Test early, release often, catch more                       | 6  |
| Scale without compromise                                    | 7  |
| Built for regulated industries                              | 8  |
| The proof is in the numbers                                 | 9  |
| Standing still isn't free                                   | 10 |
| The version of you that never gets an urgent ping at 2 a.m. | 11 |

# Applications don't fail quietly anymore

A slow API. An unpredictable AI model. A traffic spike no one planned for. Today, the outage is trending before your team even gets the alert, and by the time performance problems reach production, the damage is already done: lost revenue, eroded trust, and an engineering team stuck firefighting instead of building.

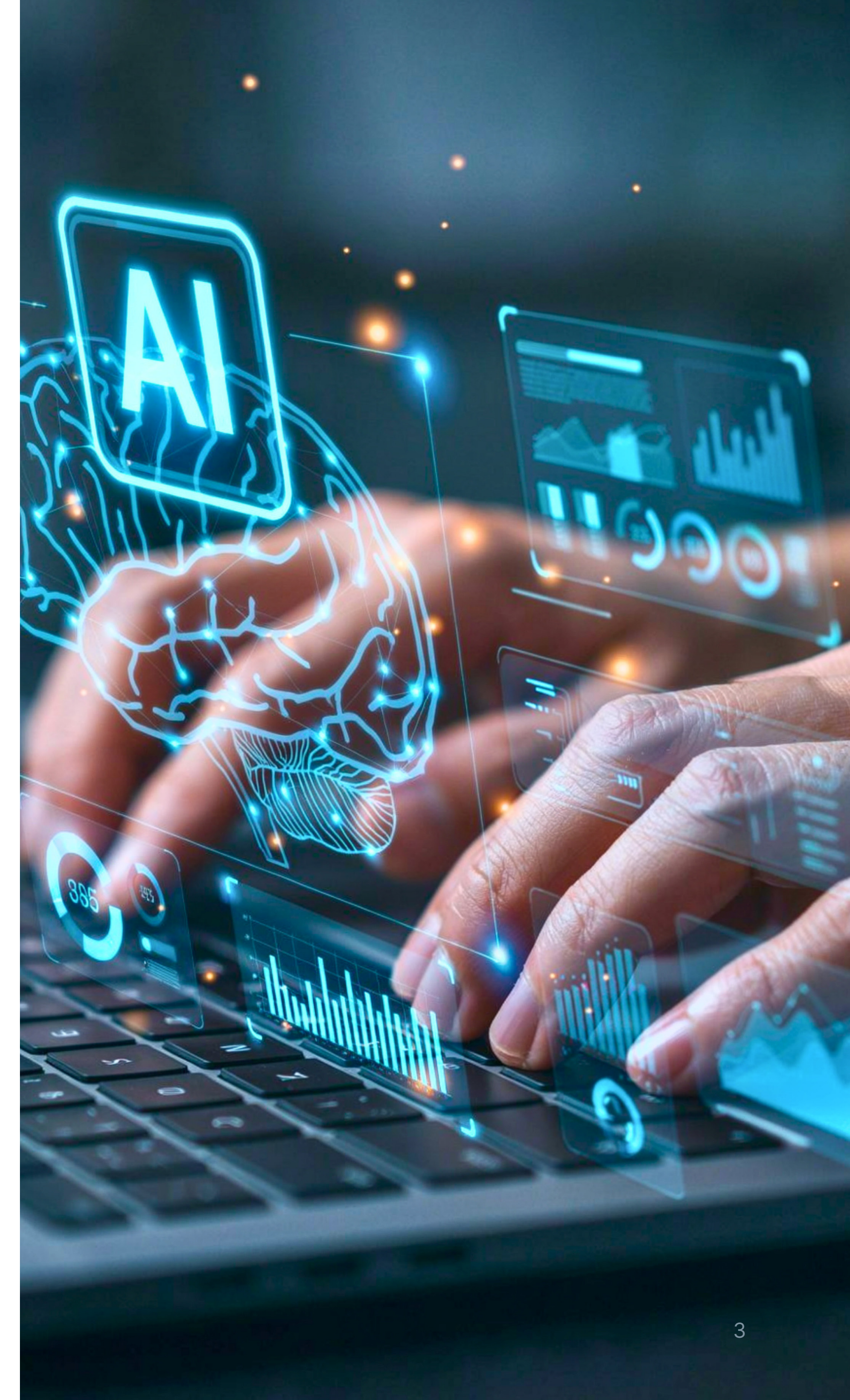
This playbook lays out how the best [performance engineering](#) teams are closing that gap:

- Shifting validation earlier.
- Removing the skill barrier holding teams back.
- Scaling testing without scaling headcount.

So every release is validated at real-world scale, not just under ideal conditions.

You'll learn how to:

- Use AI to write and optimize test scripts.
- Turn performance data into actionable insight.
- Catch root causes before performance issues ever reach the customer.
- Build these practices into the CI/CD pipeline your team already runs.

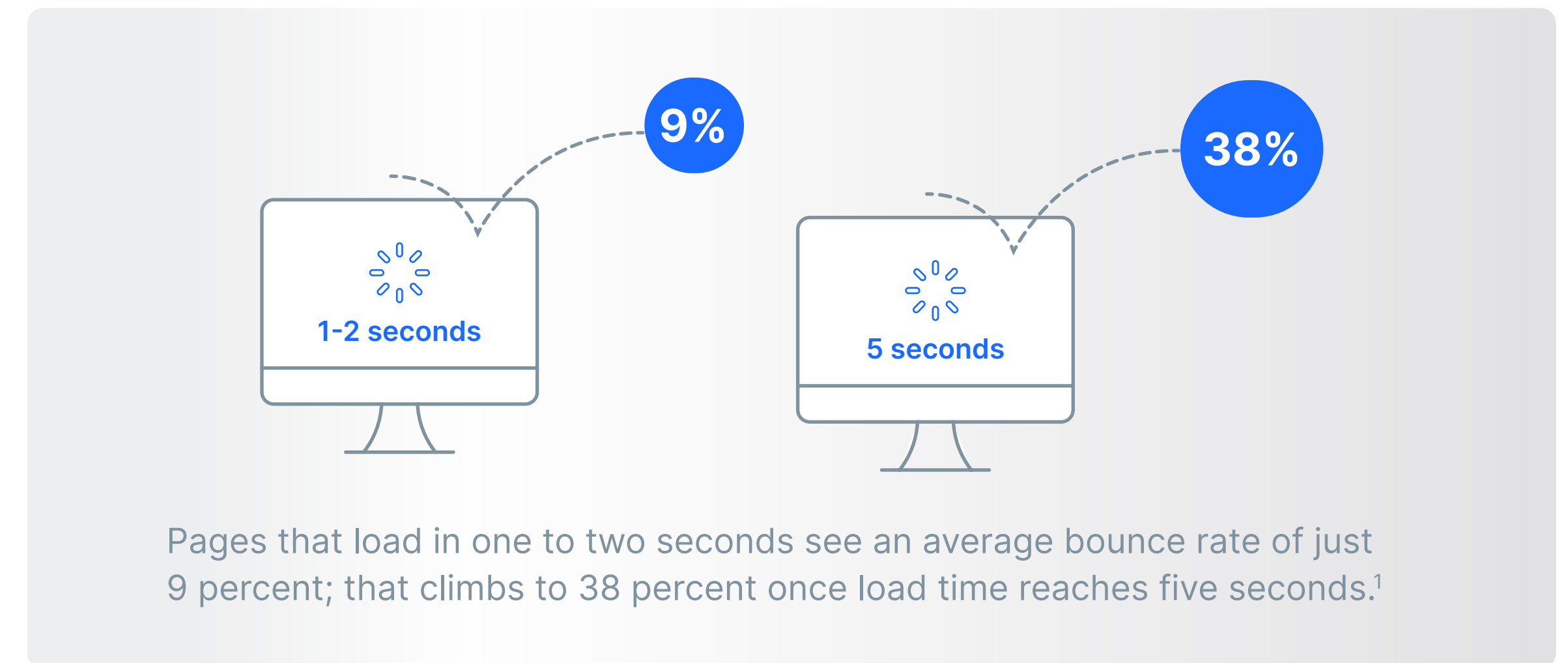


# When “it works” isn’t the same as “it holds up”

Outages and slowdowns impact more than uptime. They hit revenue, SLAs, and the trust customers place in your applications. Pages that load in one to two seconds see an average bounce rate of just 9 percent; that climbs to 38 percent once load time reaches five seconds.<sup>1</sup> For every 0.1 second slower a site loads, conversion tends to be about 3.5 percent lower.<sup>2</sup>

Meanwhile, engineering teams are stuck in reactive mode—on-call fatigue, last-minute firefighting, and pressure to release faster without the tools to validate that speed is safe. Unplanned downtime now averages \$14,056 per minute, and 93 percent of organizations report that a single hour of downtime costs more than \$300,000.<sup>3</sup>

The result is a widening gap between how fast applications are being built and how confidently they can be released. Closing that gap starts with rethinking when, and how, performance gets tested.



Unplanned downtime  
now averages

**\$14,056** per minute



**93% of organizations**

report that a single hour of  
downtime costs more than  
**\$300,000.**<sup>3</sup>

<sup>1</sup> DigitalApplied, *Page Speed Statistics 2026: Performance and Revenue Impact*

<sup>2</sup> Shopify, *Store Speed and Conversion: What the Data Shows, 2026*

<sup>3</sup> Enterprise Management Associates (EMA), via BigPanda *The Rising Cost of IT Downtime, 2024*

# Where AI does the heavy lifting

Writing and maintaining test scripts has always been one of the slowest, most manual parts of performance engineering. AI-powered performance engineering intelligence changes that. You go from idea to test in minutes by writing and [optimizing scripts automatically](#).

That same intelligence carries through to what happens after the test runs. Complex performance data gets turned into clear, actionable insight, with [accelerated bottleneck detection](#) and resolution, so teams spend less time hunting through logs for root causes and more time acting on what they find. Every result remains **governed and audit-ready**, so teams validate applications with the same confidence everything else they release gets.

This isn't AI replacing performance engineers. It's **AI clearing the two most time-consuming parts of the job**, scripting and analysis, so engineers can focus on what actually needs their judgment.

That intelligence extends to a newer challenge: AI-powered features themselves. Chatbots and smart search built on an LLM don't behave like a normal page load. Many display their answers gradually, word by word, instead of all at once, and some depend on outside providers whose speed your team doesn't control. Dedicated support for testing applications with an embedded LLM gives teams **a live view of LLM-specific speed and responsiveness**, not a blind spot discovered after launch. That same coverage extends to MCP (Model Context Protocol) workflows, so teams can **validate performance wherever AI agents connect into their applications**.

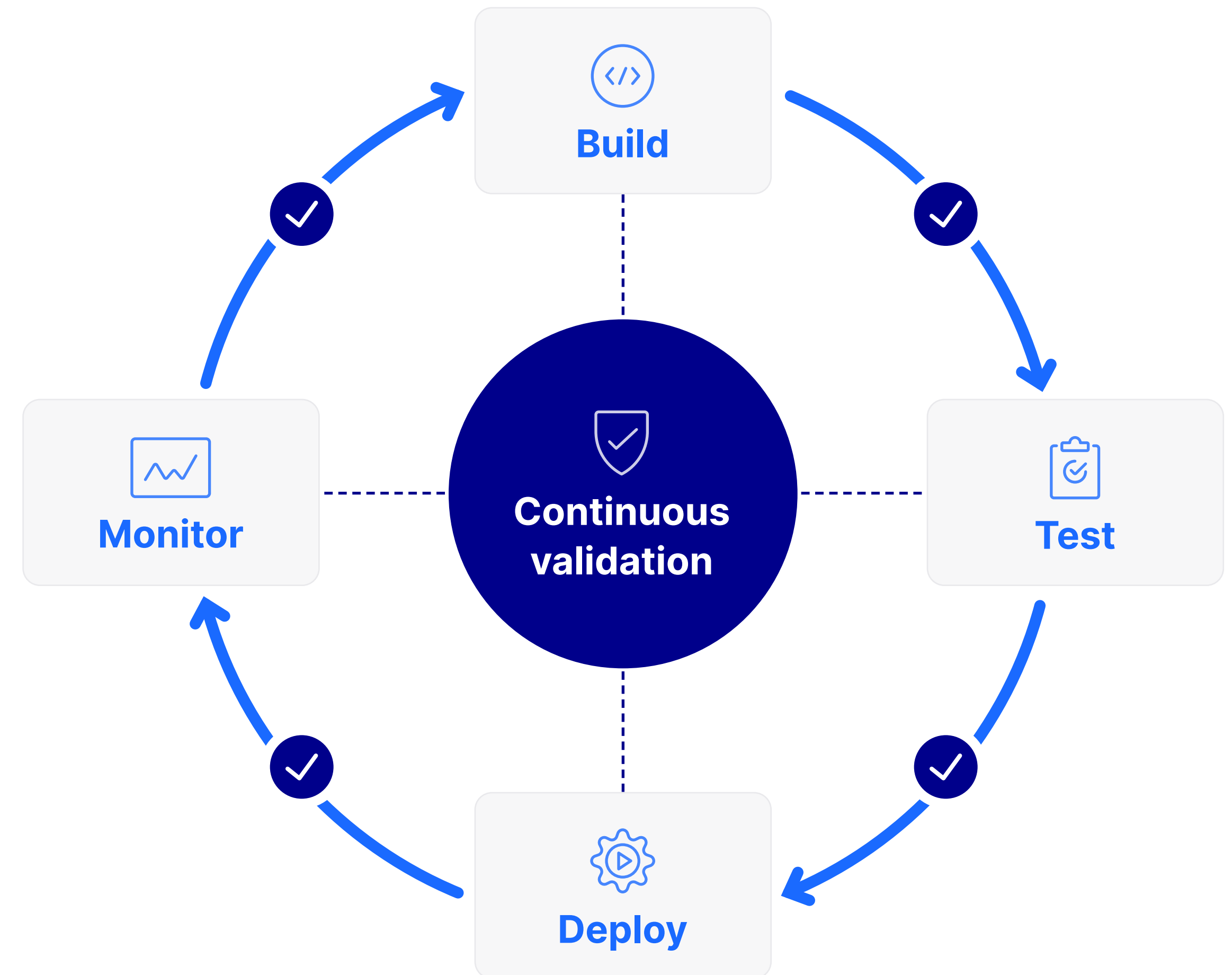


# Test early, release often, catch more

Modern software delivery moves fast, which means performance validation has to move with it. [Shift-left](#) validation and real-world load simulation surface issues while they're still cheap to fix. Defects caught during QA cost roughly 15 times more to fix than those [caught at requirements](#); once they reach production, that multiplier climbs to 30–100 times.<sup>4</sup>

[No-code, browser-based testing](#) removes the skill barrier that's historically kept performance testing limited to a handful of specialists. Any team member can design and run tests, not just the few with deep scripting expertise. And because validation runs directly inside the [CI/CD](#) toolchain, teams can keep pace with modern architecture, including microservices and distributed systems, without slowing releases down.

Every bug caught early is a crisis that never happens.



Performance validation is embedded at each CI/CD stage

<sup>4</sup> BetterQA, *Cost of Fixing Bugs by SDLC Stage: Why Production Is 30x*, 2026

# Scale without compromise

Today's application landscape isn't one thing. It's legacy systems, modern APIs, SAP and other packaged enterprise apps, LLM-powered applications, and more, all running side by side. Testing them shouldn't mean stitching together multiple tools to cover them all. What teams need is one solution built to cover the full range of protocols and technologies they're testing today, and the ones they'll need to test tomorrow.

That coverage must hold true at enterprise scale, simulating massive, real-world load with elastic, distributed execution on demand. And it must deploy anywhere: **cloud, on premises, or hybrid**, at full enterprise scale, with the governance and control every enterprise needs, including the audit readiness regulated industries can't do without.

Teams using this approach have completed **17 times more tests per week**, scaling testing without scaling headcount, and **cut performance testing time by 30 percent**, freeing budget and hours for the work that actually needs them.



A selection of the technologies and applications we support.  
This is not an exhaustive list.

# Built for regulated industries

For financial institutions, life sciences companies, and public sector organizations, [performance testing](#) isn't just an engineering concern; it's a compliance one. The EU's **Digital Operational Resilience Act (DORA)** requires financial institutions to prove documented operational resilience, including how systems hold up under real-world disruption, which is where [chaos engineering](#) comes in. Public sector organizations face similar pressure under **FedRAMP**, and life sciences companies must satisfy **FDA validation requirements**. Open-source and standalone tools weren't built to produce that evidence or the traceability regulated environments require.

Enterprise-grade performance engineering closes that gap, generating audit-ready evidence and **full traceability as a built-in part of testing**, not an afterthought bolted on before an audit.

For enterprises in regulated industries, this turns a compliance requirement into a competitive advantage: faster audit readiness, reduced regulatory risk, and stronger executive-level governance.



# The proof is in the numbers

The business case for performance engineering isn't abstract. Across organizations putting performance engineering to work, the impact shows up directly in **reliability, speed, coverage, and efficiency.**

**98%+**

**improvement in application performance**, turning peak-traffic moments into wins instead of outages

**70%+**

**faster test script creation**, so teams spend less time scripting and more time shipping with confidence

**75%**

**reduction** in debugging effort, accelerating issue resolution and improving tester productivity

**30%**

**productivity improvement** through AI-assisted troubleshooting

**90%**

**time saved** in documentation with automated test case generation

# Standing still isn't free

The cost of standing still is rising fast. Downtime costs for the world's largest companies have surged 50 percent in just two years and in a recent survey of technology leaders, every single respondent said their organization has experienced AI-related downtime, even though most believe AI should be reducing that risk, not adding to it.<sup>5</sup> That gap between expectation and reality is exactly where performance engineering earns its keep.

Every release shipped without real-world-scale validation is a release shipped on hope. The organizations pulling ahead are the ones treating performance engineering as something built into every release, not tested at the end, and not left to chance.



The Cloud Security Alert Interface actively monitors and responds to potential cyber threats in real time, offering businesses a powerful tool to safeguard sensitive information. With advanced data protection protocols, automated backup systems, and threat detection algorithms, this technology enhances the resilience of IT infrastructure. It empowers organizations to maintain continuity, minimize risk, and ensure safe digital operations in an increasingly connected world.

"Cloud-based threat alerts help protect your digital systems by detecting risks early, backing up important data automatically, and keeping your business running securely. This technology gives you peace of mind and the tools needed to operate safely online."

# The version of you that never gets an urgent ping at 2 a.m.

Applications don't fail quietly anymore—but with the right approach, your team doesn't have to find out the hard way. AI-powered, hybrid-ready performance engineering catches bottlenecks in development, not production, validates every release at real-world scale, and builds governance in so that scale never costs control.

**Ready to see what performance engineering built for the next generation of applications looks like?**

[Start releasing with confidence ›](#)

See these capabilities in action with guided click tours

→ **OpenText™ Core Performance Engineering**

Scalable, cloud-based performance testing that grows with your business

→ **OpenText™ TruClient**

Browser-based testing that's fast, accurate, and simple enough for any team member to run

→ **OpenText™ Core Performance Engineering Analysis**

AI-powered analysis that turns performance data into faster, more reliable answers.

→ **AI-powered scripting in VuGen using Performance Engineering Aviator™**

AI-powered scripting that cuts script time, removes the guesswork, and helps you fix issues faster

## About OpenText

OpenText, The Information Company, enables organizations to gain insight through market leading information management solutions, on premises or in the cloud. For more information about OpenText (NASDAQ: OTEX, TSX: OTEX) visit [opentext.com](https://opentext.com).

[opentext.com](https://opentext.com) | [X](#) | [LinkedIn](#)