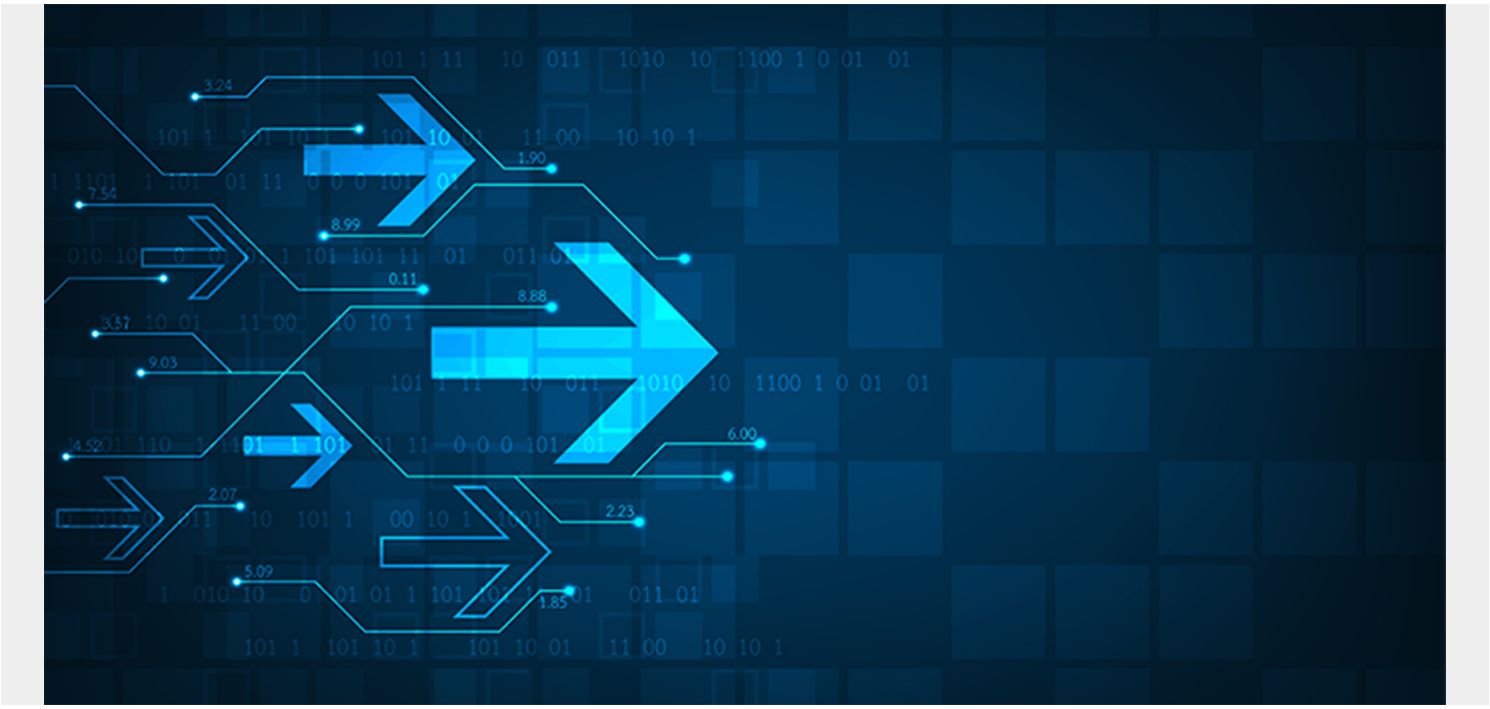
TOP CONTINUOUS IMPROVEMENT METRICS DEVOPS TEAMS SHOULD CARE ABOUT



Aligning software delivery teams around a single source of truth not only enables them to do their jobs better, it also helps them visualize what needs improving. DevOps leaders need such insights to quickly unlock value now and understand the tipping point where technical debt begins to create diminishing returns. Continuous improvement is based on the idea that what doesn't get measured doesn't get improved; but some metrics matter more than others, and too many can lead to analysis paralysis. So, what is the right mix to move the needle towards value creation?

One compelling metric from the *2025 BMC Mainframe Survey* is that mainframe workloads have grown 24 percent over the past five years. This comes from mainframe developers armed with modern tools and capabilities working towards breaking down silos and focusing on continuous improvement. And that's helping to maximize mainframe workloads, optimize costs, reduce risks, and generate high-impact value streams. Enterprise software delivery teams that analyze core DevOps key performance indicators (KPIs) can quantify the payoffs of their digital transformation investments, and get better, faster.

So, what are the core areas that DevOps leaders can focus on to drive quick value today and stay the course to achieve modernization outcomes tomorrow?

The DevOps DORA 4 metrics: the building blocks of quality

Introduced by the DevOps Research and Assessment team (DORA), the DevOps DORA 4 metrics help DevOps leaders accelerate software delivery and increase digital agility through these four areas:

- **Deployment frequency**—how often an organization successfully releases to production
- **Lead time for changes**—the amount of time it takes a commit to get into production
- **Change failure rate**—the percentage of deployments causing a failure in production
- **Time to restore service**—how long it takes an organization to recover from a failure in production

These data points are extracted from across DevOps tool chains to expose insights around each phase of the DevOps delivery cycle. Smaller companies running workloads on both mainframe and cloud may have an easier time visualizing the DevOps DORA 4. But enterprises with a vast portfolio of back and front office software applications to manage are often more complex and have greater efficiencies to gain by optimizing around these KPIs.

The trifecta of continuous improvement

Velocity

Velocity is the right balance between speed and accuracy. It exists in harmony with, not at the expense of, quality and efficiency. Velocity data is mined from code commits, remediations, deployments, down time, and burndown rates derived from source code management (SCM), testing, integrated development environments (IDEs), IT service management (ITSM), and development tools like the BMC AMI DevX tool suite.

Continuously benchmarking progress is vital to achieve results and make informed decisions. Velocity KPIs include development lifecycle time and change failure rate, as well as:

- **Deployment frequency**—how often is code deployed to production
- **Mean time from checkout to production**—how much time the entire lifecycle takes, from when code is checkout by the developer to when it is deployed to production

Quality

Swift delivery means nothing if the product does not meet quality standards. Comparing trends of hot fixes and rollbacks, delivery teams can rationalize change failure rates. Whether failure rates are increasing or declining, defect density shows how many bugs are escaping into production environments in near-real time. A decreasing defect density indicates improved code quality and fewer post-production issues.

Quality KPIs:

- **Escaped bug ratio**—the ratio of bugs that occur in production versus all environments
- **Change failure rate**—the percentage of production deployments that result in failure

Efficiency

Shift-left development approaches test sooner and more often, leaving developers with more data to analyze more frequently. Efficiency metrics examine defect ratios, developer productivity, uptime, product usage, and cost optimization factors over time. This helps chart progress toward shipping faster, with greater frequency and fewer bugs. Automated testing can greatly improve efficiency metrics, and for organizations still doing manual testing, efficiency measures can substantially

reduce costs and risks to help increase competitive posture.

Efficiency KPIs:

- **Lead time for change**—determines the time required to deploy new releases after the developer has implemented changes to the code
- **Innovation percentage**—the amount of time that developers spend on activity that results in new functionality being deployed to production versus the amount of time they spend on bug fixes

Aligning around a common data-driven vision

Mainframe teams and data can't exist in silos. Democratizing insights around a common source of truth empowers the entire software delivery team—from leadership to developers—with the insights to make the right decisions. While platforms are fundamentally different, DevOps teams should be aligned around common KPIs, tools, and workflows. This prioritizes the mainframe's role in unlocking digital transformation goals, even in cautionary times.

Measuring testing itself

Some organizations may still be using old tests to monitor performance on refactored applications. Unfortunately, it's not sufficient to simply track incremental improvements or declines whenever testing takes too long or when it delivers inconclusive results. Long response times have critical impacts on the customer experience, so it is vital to conduct performance testing at the system level to evaluate response times, CPU and memory utilization, and I/O rates. When a single minute of delay can result in the loss of a customer, it is mandatory to track trends and see problems as they arise, before they escalate to a bigger problem.

The future of mainframe monitoring is in AI/ML

If developers are manually reviewing log files, code commits, and test results, then they are not writing code. Developers are happiest when they're writing code. Artificial intelligence and machine learning (AI/ML) offer new and easier ways to enhance and simplify continuous improvement. In a recent podcast, BMC Vice President of Research and Development Dave Jeffries commented that natural language processing (NLP) helps developers formulate the right questions, and get to the right answers faster. By analyzing patterns to determine what normal looks like, predictive analytics tools like BMC AMI zAdviser help teams align around the right corrective actions and understand what good looks like. While "trust but verify" may still be the golden rule of AI today, developers can achieve success guided by AI/ML-led continuous software delivery insights.

Learn more about continuous improvement

If you want to dig deeper into continuous improvement on the mainframe, watch the webinar, "Driving DevOps and AIOps Continuous Improvement on Mainframe," to learn more about how AI/ML is enabling faster DevOps through cross-team analytics.