Charting Your Mainframe DevOps KPIs

A BMC Compuware Guide to Metrics Worth Measuring
Compuware joins BMC to empower the next generation of developers to mainstream the mainframe. With unparalleled mainframe operations management and agile application development and delivery, BMC Compuware provides a mainframe-inclusive DevOps toolchain that increases volume and velocity.

Bring apps that run on the mainframe to market faster with increased quality throughout the process of analyzing, building, testing, deploying, monitoring, and tuning apps and services. Our combined offering accelerates each client’s evolution to an **Autonomous Digital Enterprise**.
Introduction

DevOps is proving indispensable for large mainframe-enabled organizations battling disruption. It’s helping recharge lines of communication, inspire collaboration, and re-establish transparency. It’s helping accelerate mainframe application development and delivery with modern processes and technology. And it’s helping better align mainframe environments with the engagement technologies they support so you can streamline the delivery of innovation to customers.

But as mainframe workloads grow and experts retire, your teams must transition from “doing DevOps” and “doing Agile” to continuously improving their development and delivery quality, velocity, and efficiency under the evolving culture, processes, and toolchains they’re being empowered with.
KPIs: Your Route to Continuous Improvement

To continuously improve your DevOps efforts, you need to set objectives for eliminating bottlenecks, waste, and inefficiencies in your mainframe environment that are unique to it. You meet these objectives by establishing key performance indicators (KPIs) and metrics that enable you to measure progress and make data-driven decisions towards success.

The KPIs your organization measures will differ from what other organizations measure. However, research BMC Compuware conducted with Forrester shows all organizations should be measuring three things in unison without a doubt:

- Quality
- Velocity
- Efficiency

We strongly believe engagement is another area worth measuring, as it affects each of the other three KPIs. To measure productivity across these areas, we offer active BMC Compuware customers a free service: **BMC Compuware zAdviser**. It uses machine learning to find correlations between developer behaviors and KPIs based on data you provide from development efforts, helping you make data-driven decisions to continuously improve quality, velocity, and efficiency.
Too much focus on one objective will decrease the effectiveness of the others, so striking a delicate balance between each and continuously adjusting them as bottlenecks move is imperative to making real improvements in developing and delivering competitively differentiating innovation.

In this paper, rather than giving you a litany of metrics to start measuring, we’ll share some basic but powerful knowledge based on leading research as well as our own experiences and those of our customers. Get ready to learn about:

**KPI Objectives**
- to focus on quality, velocity, and efficiency

**Metrics Worth Measuring**
- to help you gauge your progress

**Continuous Improvement Goals**
- with new practices for improving metrics

**Continuous Improvement Enablers**
- to support the new practices you implement

You will see there are constant and clear overlaps between each KPI, as there should be—each impacts the other indefinitely. For the sake of brevity, we won’t dive into the nuances between KPIs and metrics or the various practices and tools you can leverage to simultaneously improve each. We think these commonalities will become obvious on their own. Refraining from making tangential connections for you will give you an opportunity to judge for yourself how to apply the information we provide here with flexibility in your own environments.

Let’s get started.
KPI OBJECTIVE #1

Increase Development Quality

Given the high volume of complex business transactions mainframes process every day, accuracy is and always has been a priority. Mainframe professionals have always paid meticulous attention to quality, spending as much time as necessary to ensure relatively bug-free applications. And as mainframe organizations are asked to deliver code faster to support digital transformation, measuring quality becomes even more important.

But a myopic focus on quality will become a constraint to velocity and efficiency. As your mainframe teams coalesce with non-mainframe teams under DevOps, it’s critical they continue their success with quality. However, they must reduce the time and increase the efficiency with which they ensure it.
Escaped vs. Trapped Defects: Continuous Improvement Goal #1

Increase Program Understanding: Provide a means for developers to visualize programs and identify potential code-change impacts so they can make intelligent decisions. Publish coding standards that serve as guidelines developers can follow to ensure consistency.

Escaped vs. Trapped Defects: Continuous Improvement Enablers for Goal #1

**BMC COMPUWARE TOPAZ**

BMC Compuware Topaz for Program Analysis for the visualization of unfamiliar or complex mainframe applications—including runtime analysis with its Runtime Visualizer feature and static code analysis—to improve developer understanding before making changes.

**BMC COMPUWARE ISPW**

ISPW to show the potential impacts of code changes during builds, to compare versions of code, to spot and easily merge changes, and to automatically regress problematic versions to the best, most recent version.

**SONARSOURCE SONARQUBE**

SonarQube for tracking program defects, complexity, and technical debt while displaying results in an intuitive dashboard that eases management of projects.

**SONARSOURCE SONARLINT**

SonarLint for on-the-fly code-quality checks and issue flagging for easier troubleshooting.
Escaped vs. Trapped Defects: Continuous Improvement Goal #2

**Automate Testing:** Automate your repetitive, error-prone, often-bypassed manual testing practices that let defects slip into production and prolong development cycles. Building out a repeatable testbed increases your chances of trapping defects, and increasing tests through automation over time makes it easier to identify recurring issues that could signify poor coding practices causing the issues. Overall, automated testing improves code quality and frees up more time for innovation. The real benefit is improved customer satisfaction, as less defects will be discovered by them.

**Escaped vs. Trapped Defects:** Continuous Improvement Enablers for Goal #2

**BMC COMPUWARE TOPAZ FOR TOTAL TEST**

*Topaz for Total Test* for automating the creation and execution of unit tests to find and fix bugs earlier in the development process, before testing or production.

**BMC COMPUWARE HIPERSTATION**

*Hiperstation* for repeatable testing through the creation of automatable scripted tests that replay functions in unattended mode so you don’t overlook issues.
Escaped vs. Trapped Defects: Continuous Improvement Goal #3

**Increase Debugging with Code Coverage:** Data shows effective triage teams that maintain high quality increase debugging as defects increase. Increase your debugging to find and fix bugs ahead of deployment. Use code coverage to see that code has been tested and is executing correctly to prevent bugs from escaping. Implement quality gates to prevent low-quality code from moving into the next phase.

**Escaped vs. Trapped Defects: Continuous Improvement Enablers for Goal #3**

**BMC COMPUWARE XPEDITER**

Xpediter for interactive debugging and Xpediter Code Coverage for a visual representation of captured code-execution statistics that show if new code is being tested and meets quality standards.

**SONARSOURCE SONARQUBE**

SonarQube for quality gates and for code coverage to compare code execution results and the percentage of an application that has been tested with broader cross-platform quality trends and metrics.

*A myopic focus on quality will become a constraint to velocity and efficiency.*
KPI OBJECTIVE #2

Increase Development Velocity

As mainframe teams work to accelerate development and delivery with DevOps, measuring that velocity helps determine how much work can be completed within a given timeframe. Another way to look at it is measuring how fast mainframe teams can react to deliver a requirement.

Much of mainframe development revolves around maintenance and fixing bugs, leaving little time for true innovation. In large part, delivering more code faster will require finding ways to find and fix bugs faster.

One area that is affected by velocity is how frequently you can deploy new innovations and fixes into production. DevOps practices make continuous deployment possible. Using short, quick feedback loops, you can deploy as often as several times a day. The more efficient your team is, the quicker you can deploy changes to your customers.
MTTR and MTTD: Continuous Improvement Goals

“Shift Left” Error Detection: Make it easier to quickly pinpoint issues earlier in the development process with intelligent error detection and visualization of error trends and spikes. Integrate detection into an issue-reporting tool that tracks and prioritizes bugs in a backlog, helping developers tackle the biggest issues first.

Enable Interactive Debugging: Developers tasked with issue resolution should be able to quickly start a “step-through” debugging session with line-by-line visibility of code execution, helping them quickly pinpoint and validate issues to fix.

Automate Unit Testing: Accelerate the process of finding low-level bugs in the small parts of applications early in the development process so the issues in question are easier and faster to fix. This also ensures cleaner code moves through subsequent testing phases, helping them go faster.

Implement Unit Test Coverage: Code coverage of unit tests shows what code has been executed and what percentage of an application has been tested, reducing the time needed to get programs implemented or returned to production by helping developers more quickly determine where code needs attention.

MTTR and MTTD: Continuous Improvement Enablers

BMC COMPUWARE STROBE

Strobe for application performance monitoring and analysis to detect performance problems in production that can generate issues in Atlassian Jira to be tasked to a developer.
BMC COMPUWARE ABEND-AID

Abend-AID for application failure resolution and fault management to “shift left” error detection and visualize failure trends and spikes through an integration with Precisely Ironstream into Splunk or the Elastic ELK Stack.

BMC COMPUWARE XPEDITER

Xpediter for fast, interactive debugging line by line.

BMC COMPUWARE TOPAZ FOR TOTAL TEST

Topaz for Total Test for the automated creation of unit test scenarios, test cases, test assertions and test data to eliminate slow, repetitive, manual processes and accelerate bug fixes.

BMC COMPUWARE XPEDITER CODE COVERAGE AND SONAR SOURCE SONARQUBE

Xpediter Code Coverage and SonarQube for visual representations of captured code-execution statistics for faster determination of whether statements are executing as expected before moving forward.

Features/Modules Deployed: Continuous Improvement Goals

Shorten Development Lifecycles: Establish a faster delivery cadence by reducing requirements and shortening development lifecycles to iterative two-week sprints through which you deliver minimum viable products: specific pieces of functionality delivered in smaller batches to satisfy immediate customer needs based on their feedback. Deploying features containing fewer modules lowers your risk of introducing something bad.
Enable Side-by-side Compare: Make it easy for developers to compare code versions, spot changes in code, and merge code changes faster before deploying those changes.

Orchestrate Releases: Find a way to automate, standardize, and monitor code deployments across multiple platforms into multiple target environments to ensure mainframe and non-mainframe code can deploy at the same rate when necessary.

**Features/Modules Deployed: Continuous Improvement Enablers**

**BMC COMPUWARE ISPW**

ISPW for Agile source code management, enabling concurrent development, easing communication, and collaboration through features like code check-out alerts, and improving visibility of program versions throughout the development cycle to ensure smooth code integration.

**DIGITAL.AI**

Digital.ai XL Release for orchestrating the automation, standardization, and monitoring of code deployments across multiple platforms into multiple target environments.

**JENKINS**

Jenkins for accelerating the software development process through the orchestration and automation of the deployment pipeline.
KPI OBJECTIVE #3
Build Efficiency

Measuring efficiency helps you understand how successfully work is executed without waste. In another sense, measuring efficiency is how you determine where your biggest inefficiencies are hiding—including some that have existed for years as byproducts of a previous waterfall environment.

This is a major part of analyzing your deployment pipeline to see how you can improve your DevOps processes as well as help your teams improve their usage of tools to ensure the technology they have at their fingertips is being leveraged appropriately and to the fullest extent.
Lead Time: Continuous Improvement Goals

Strategize Ideation with Mainframe Teams: Ensure mainframe teams are continuing to be included in the use of Agile methodologies like Scrum that bring people from various roles together to converge, discuss, and review ideas with cross-functional insight. Scrum meetings include standups, sprint planning, sprint reviews and sprint retrospectives, and product backlog refinements. Each phase helps facilitate a disciplined, organized path to prioritizing the most important ideas to work on, ensuring they’re acted upon quickly and ensuring they’re monitored in production.

Analyze Your Lifecycle: Track how many days it takes from the time changes are put in to the time a feature launches. That is, analyze your deployment pipeline to find latency slowing the development of your ideas. This could be due to poor idea prioritization, too much manual testing, too many manual handoffs, lack of program visibility, lack of debugging, or lack of code coverage.

Improve Coding Practices: Where you find latency is a sign of where you should improve coding practices. There should be a process for prioritizing ideas. Repetitive, manual processes that can be automated should be. Visibility of the program and its journey through the pipeline should both be available. Ramp up debugging and code coverage to find and fix bugs earlier.

Lead Time: Continuous Improvement Enablers

ITSM solutions like Jira that allow you to set measurable statuses for ideas—such as notifications of when development is complete or when it’s time to review code—to understand what steps are being taken, and when, for ideas to move forward.
BMC COMPUWARE TOPAZ WORKBENCH

Topaz Workbench to use your core development and testing products in an Eclipse-based IDE (interactive development environment) that integrates with a range of leading DevOps solutions and displays a single, common, intuitive interface to enhance the productivity of novice and expert developers.

Percentage of Time Spent: Continuous Improvement Goal

Innovate, Innovate, Innovate! To get as much innovation as possible through the pipeline, you first need to understand what your development teams are working on. Are they working on bugs, new innovation, or sitting in meetings all day? Until you start collecting and analyzing this data, you won’t know. Then you need to remove impediments to create more time for new development.

Some things can’t be reduced or eliminated, e.g., Agile ceremonies, company-wide town halls, and employee vacation time. The solution, then, is to automate what you can in the pipeline to remove the need for human intervention, which amounts to time not spent innovating. As you’ve noticed by now, this means automating testing, code coverage, deployment, and more where it makes sense.

Percentage of Time Spent: Continuous Improvement Enablers

Automation tools that span your build, test, and deploy phases, including, but not at all limited to, many of those already referenced in this paper: the Topaz suite, ISPW, XL Release, Electric Cloud, Jenkins, SonarQube and more.
Employee engagement, which fuels a company’s culture, has been greatly overlooked at many large organizations—but it greatly impacts your development quality, velocity, and efficiency, and it is essential to a flourishing DevOps ecosystem. Understanding how you can convert unengaged and disengaged employees into passionate explorers will place distance between you and your competitors because your passionate explorers are the ones who drive innovation.

Measuring engagement can come off as invasive and highly controversial at first, but when only 33 percent* of employees are engaged (they crave challenges and strive for more than doing a decent job), 51 percent are unengaged (they show up, they go home), and 16 percent are disengaged (they resent work, resent the company, and are actively working against your success), figuring out how to close the gap between the engaged, unengaged, and disengaged is imperative to your organization’s success—while working to nurture passionate explorers who lead the way in DevOps transformations.

Surreptitiously measuring engagement is antithetical to DevOps transparency, so ask employees to anonymously complete annual surveys that capture their feedback across a spectrum of areas related to their satisfaction with your organization, asking questions like:

+ Do you understand where our industry is headed in the next five years?
+ Do you believe the ideas our organization has delivered (or is delivering) are making a competitive difference in the eyes of our customers and potential customers?
+ Do you feel empowered to take appropriate risks to make our organization successful?

With this feedback, you can begin to investigate the overall levels of engagement throughout your teams and organization, without calling out individuals. And by pairing this feedback with the data you capture through other measurements in quality, velocity, and efficiency, telling patterns may emerge that help you understand where pockets of the unengaged or disengaged exist in your teams and organization.
Make Data-driven Decisions

DevOps transformation is a big job for large, tightly coupled organizations. You must define clear business objectives and measure how changes you make are helping you meet your goals. Figuring out those business objectives should entail executives taking time to determine what is and isn't working in the organization, and then prioritizing the removal of impediments.

The process for understanding your objectives should start in the deployment pipeline, as it illustrates how business value moves through IT to the market. It’s where you can develop an understanding of where bottlenecks and waste exist in your DevOps processes for building, testing, and releasing large enterprise systems.

And remember, your end state is never “achieved”—your end state is one of continuous improvement away from the status quo. Aligning with the right KPIs and measuring the right metrics is the only way to truly sustain that journey and experience real DevOps progress.

BMC COMPUWARE ZADVISER

Using a machine-learning service like zAdviser will make this process substantially smoother by giving you a new perspective of how your current tools and processes are being leveraged for quality, velocity, and efficiency.

Additional Resources

- Starting and Scaling DevOps in the Enterprise by Gary Gruver
- “Modern Mainframe KPIs Are Key to a Successful Digital Strategy” by Forrester
- “The Need For Speed: Drive Velocity And Quality With DevOps” by Forrester
- “Use Four Key Categories To Measure What Matters In Continuous Deployment” by Forrester
About BMC
From core to cloud to edge, BMC delivers the software and services that enable over 10,000 global customers, including 84% of the Forbes Global 100, to thrive in their ongoing evolution to an Autonomous Digital Enterprise.

BMC—Run and Reinvent  www.bmc.com