

Real World DevOps for Mainframe Enterprises

How Smarter Mainframes Achieve DevOps Levels of Efficiency

Table of Contents	page
Rationale	2
About This White Paper	2
About DevOps	2
What's in a Name?	2
DevOps Defined	3
Sounds Great...	3
So What's So Hard About It?	3
The Way Ahead?	4
Scenario A: Improving the Effectiveness of Parallel Development for Multiple Teams	4
Scenario B: Improving Developer Productivity	5
Scenario C: Fast Provisioning and Test Environment Delivery	6
In Conclusion	7

Rationale

Our industry has the fastest-changing landscape of any contemporary industry: Discuss. Our industry is the home of more nebulous and potentially empty phrases and terminology than any other: Case closed.

Unusually, we will use the platform of a DevOps white paper to challenge the concept of DevOps itself. We will deconstruct its definition—such as it is—and instead reveal a framework of best practice, achievable tweaks to current working practices and, crucially, explain how using the right tooling can deliver all the supposed benefits of DevOps without the tenuous nomenclature.

About This White Paper

We reflect the ethos of DevOps at work in the real world by citing three genuine scenarios of companies with infrastructures and technology profiles far removed from the normal DevOps case study. A bank, a large insurer and an SI are all good examples of enterprises implementing DevOps-style improvements and enjoying better business outcomes as a direct result.

This paper is a deep dive into what DevOps *really* means and how organizations can achieve the principles of improvement away from the constraints of DevOps.

Because our philosophy is that moving away from previous inefficiencies will look different for each company, but everyone can do it.

Adopting DevOps is not simply a question of flicking a switch. It is a matter of discovering what changes will fit the profile of each organization and implementing the technological or procedural changes that will deliver the desired improvements.

This paper moves from thought-leadership towards a practical guide to improving development practices. It explains them using 'real world' cases of enterprises implementing the DevOps philosophy to improve business efficiency. And it works.

Think of this white paper as less of 'how to do DevOps' and more of an opportunity to identify and correct areas of development improvement.

About DevOps

The technology world has changed almost immeasurably in the last couple of decades. Platforms, architectures, software, communications methods, have all undergone seismic shifts such that the technology world is entirely unrecognizable from previous generations.

Not only has the output of technology innovation changed, but so too has the process by which that technology is created.

It has also spawned a new generation of terminology that inspires some and creates an existential fear of missing out—or being left behind—in others.

Perhaps there is less to DevOps than meets the eye; at its core it is simply a new way of framing the best of modern working practices. Because gone are the days of a regimented, front-end-loaded, bureaucratic waterfall model, with projects measured in months if not years. In many cases, this has been displaced with a more nimble, agile process of establishing requirements and getting on with the process of building and delivering in a matter of days. Nowadays, most technology staff operate in an agile framework—or know someone who does—whether they realize it or not.

What's in a Name?

The lexicon of technological buzzwords grew by one at a 2008 Agile conference when the term 'DevOps' was spawned as a means of defining how agile methods might interoperate with the wider IT infrastructure operation. The phrase quickly gained traction to the point where today it seems to be another emerging de-facto standard in the technology world, whereas in reality it simply defines the buffer between working practices and available technology. It may be aspirational but that doesn't mean that every organization cannot implement some aspect of DevOps in what they do.

According to Wikipedia¹, DevOps is “a portmanteau of ‘development’ and ‘operations’” and is “a software development method that stresses communications, collaboration, integration, automation and measurement of cooperation between software developers and other IT professionals.”

We will look at how the principles of DevOps—the move towards a more agile methodology—can be implemented anywhere the opportunity to make improvements exists.

DevOps Defined

A recent guest article² in *SDTimes* neatly outlined the aspiration of DevOps from process, team and organizational perspectives.

- “At the technical process level, we can understand DevOps the way it got started, in the context of continuous delivery. A project is accustomed to deploying big batch releases once every quarter or so. DevOps is the aspiration to achieve much more frequent deployments.”
- “At the team level, we can understand it as an emphasis on silo-smashing IT collaboration. Our developers, testers and operations teams need to work more closely together, day-to-day, in the flow of the projects.”
- “From an organizational perspective, DevOps improvement is best understood through the Lean perspective: emphasis should be on measurement and continuous improvement through things like the identification and removal of constraints on system flow.”

In terms of tangible benefits, the possibilities seem compelling. One publication³ suggests that incorporating DevOps practices get more done, deploying code up to 30 times more frequently than their competition. And less than 50 percent of their deployments fail according to Puppet Labs 2013 State of DevOps⁴ survey.

Elsewhere⁵, other benefits are cited:

- Decreased development and operations cost
- Shorter Development Cycle
- Increased Release Velocity
- Improved Defect Detection
- Reduced Deployment Failures and Rollbacks
- Reduced Time to Recover upon Failure.

Sounds Great...

Given the promised benefits, one might confidently have predicted a litany of success stories and glowing references. However, if the difficulty in finding any online commercial reference stories is anything to go by, one could be forgiven for fearing that this was a lot of talk and not a lot of tangible outcomes. Indeed, confessions abound⁶; “enterprise DevOps is hard—but not impossible.”

Possibly one of the problems is the intangible nature of what DevOps really is. How can an organization know whether they have ‘achieved DevOps’? There is no formal accreditation. No certificate to hang up in the reception area.

The point here is that DevOps is a way of working; about achieving collaboration across the development, operations and business areas. If DevOps can really be anything tangible, it is the removal of silos, the engagement of appropriate stakeholders, the agile delivery of automated business outcomes. This is not, of itself, an automation, cloud or virtualization issue. It is about doing things better with what you have and adding appropriate technology where the process needs support.

So What’s So Hard About It?

Perhaps we need to look again about the potential issues—after all, it might be that DevOps is quite hard to do. Why might that be?

DevOps encourages a level of immediacy and collaboration across IT that makes a lot of sense in principle but is by no means straightforward to implement. This can be particularly true in long-standing enterprises. It is typically a mixture of cultural, task-related and technological issues.

- **Culture:** DevOps pre-requires a level of agility (small a, or big A, you choose) which can be a forlorn hope in many enterprise IT shops. At a recent customer event among enterprise COBOL

1 www.wikipedia.org/wiki/DevOps

2 www.sdtimes.com/guest-view-devops-is-many-things-but-theyre-all-about-building-the-right-product/

3 www.newrelic.com/devops/benefits-of-devops

4 www.puppetlabs.com/wp-content/uploads/2013/03/2013-state-of-devops-report.pdf

5 www.logicworks.net/blog/2014/10/measurable-important-benefits-devops/

6 www.informationweek.com/software/enterprise-applications/how-devops-benefits-large-enterprises/a/d-id/1268924?page_number=2

developers⁷, a cursory show-of-hands question about the use of 'agile methodologies' was met positively by less than a quarter of respondents. This ball-park level of adoption was consistent across a number of events across North America. What this tells us is that a 'typical' long-standing enterprise IT development team, namely those building COBOL applications, are not using the building blocks of a streamlined development operation, upon which DevOps arguably relies.

- **Task:** In such organizations, the reality facing development directors is the vast proportion of activities focused on regular and routine maintenance or modest enhancement. Typical estimates on the proportion of IT spend being directed towards 'lights on' activities are in the 70–80 percent range⁸. As a direct result, most development activity is not 'new', 'blank sheet of paper' type work; it requires in-depth analysis of current code, extensive debugging and unit testing and a significant investment in integration and system testing of resultant changes. Much of this has largely fixed and sequential workflows, which—especially with restrictions on technology usage—are hard to undertake any faster.
- **Technological:** Typical technological and architectural investments in larger scale enterprises often fail to keep pace with the levels of investment in other areas. As such, tooling from the 1970s and '80s is still being used as the platform for systems development. These might have been perfectly useful at the time, but they are some way short of what is possible in terms of modern development efficiency and cross-group collaboration.

The Way Ahead?

The reality with DevOps is that it promises much as a conceptual model, but requires a lot from current frameworks and underpinnings without which its implementation is jeopardized.

However, DevOps should not be seen as an all-or-nothing issue. Most of the fundamental principles driving DevOps adopting are sound, such as improved visibility, better collaboration, testing earlier in the cycle, faster delivery period.

So, let us assess the validity of DevOps in a real world context. The following three scenarios offer that level of insight.

⁷ <http://online.microfocus.com/devday>

⁸ Sources: Gartner, Forrester, Micro Focus and others

⁹ This is not a limitation of the technology per se, more a consequence of how this customer deployed it

Scenario A: Improving the Effectiveness of Parallel Development for Multiple Teams

Delivering more concurrent projects by integrating modern distributed source control to the mainframe.

Customer Profile:

A large APAC Insurance and Banking Service supporting multiple brands and many lines of business

The Challenge:

To increase delivery of new functionality to the business and enable multiple development teams to work on parallel projects.

Like many providers, this customer single insurance application provides the core business functions, but each line of business offers separate offerings for customers—and has individual demands for new functionality. A dramatic increase in the number of projects was inevitable.

Their mainframe Configuration Management System a popular mainframe source control and configuration management system, could not easily support parallel development and lacked the flexibility needed to increase the number of releases⁹. The result was a delivery bottleneck, with time being spent:

- Reproducing builds and resolving regressions
- Delivering frequent releases and
- Maintaining simultaneous lines of code
- Backing out changes and resolving merge conflicts

The Resolution:

The flexibility of distributed source control systems and modern agile development processes would enable:

- Increased developer productivity through automation
- Flexible and more frequent releases to accommodate changes
- Support for parallel development across features, releases, or teams
- Compliance adherence with full traceability

Importantly, this would be achieved with direct integration into Configuration Management on the host, ensuring a consistent promotional path to production libraries. After looking at an Open Source option, the customer chose Micro Focus® AccuRev® and Enterprise Developer for z Systems.

AccuRev¹⁰ enabled the effective support of parallel development streams, synchronized with changes to their main code libraries on the mainframe. **Enterprise Developer** enabled the synchronization of source code on the host.

The Result:

More flexibility to support parallel development streams and give developers and development teams the freedom to branch code where needed and provide process automation including promotion and auto-merging of changes all kept in-sync with the host SCM system. More projects running concurrently, delivering more functionality to the business with less time spent fixing regressions or backing out changes.

Minimized change management and adoption training effort as the AccuRev inheritance model reflects the concatenation model of mainframe development. This facilitated strings of libraries within a staged development model (DEV/INT/SYS/PROD).

Enterprise Developer enables modern Eclipse-based development, either on the mainframe or on a distributed system. This process is tightly integrated into the intuitive and uncomplicated AccuRev source configuration model.

How This Reflects the DevOps Model:

The outcomes in this scenario supported genuine IT objectives:

- **Decreased development and operations cost**—incomplete integration equals inadequate developer adoption and increased productivity costs. A more modern, efficient IDE links the developer with trusted, working mainframe processes and technology and reduces costs.
- **Shortening the Development Cycle**—exploiting workstation power to shorten mainframe development and unit test cycles, means that new products and features are made available faster
- **Increasing Application Release Velocity**—Increasing application development productivity through best-in-class developer tooling in the latest IDE technology improves output by as much as 40 percent.
- **Improving Defect Detection**—removing the bottlenecks reduces the risk to delivery schedules caused by quality, cost and time challenges
- **Reduced Deployment Failures and Rollbacks**—enabling mainframe IT organizations to perform pre-production testing on low cost commodity hardware improves success rates and avoids unnecessary cost and delay.

- **Reduced time to recover upon failure**—streamlining the development process, includes enabling 'parallel development and a faster process for getting hot fixes into production.

Scenario B: Improving Developer Productivity

Removing reliance on mainframe development resources enables flexibility.

Customer Profile:

This Micro Focus customer is a systems integrator providing mainframe development services to large corporations.

The Challenge:

To differentiate themselves from their competition, they wanted to provide a service that removed the reliance on the client's mainframe development resources. Consuming fewer development MIPS would be more cost-effective for the client, and liberate the Systems Integrator from delivery bottlenecks caused by inefficient tools and mainframe wait times.

Enabling the mainframe application development skills within the organization to work together to maximum efficiency was another problem.

The quality of talent was less of an issue than the processes they were being asked to work with. JAVA-trained and Agile-friendly developers were unsure how to approach the mainframe-based methodologies already in place.

The key was to adapt the host environment to the abilities and skillset of the incoming talent. Eclipse-based development tooling, integrated with continuous code quality management tools such as SONAR, enabled JAVA developers to enjoy an easy transition.

The Solution:

The solution was as simple as it was clever: bring Agile development to the mainframe world by enabling Java-savvy developers to connect with proven mainframe processes. Integrating Enterprise Analyzer and Enterprise Developer for z Systems into the JAVA-based development lifecycle opened up established mainframe best practice to a new breed of Agile developers.

10 A software configuration management tool with a stream-based architecture. AccuRev addresses complex parallel and distributed development environments, accelerates the development process and improves asset reuse. Source: www.borland.com/Products/Change-Management/AccuRev

Enterprise Developer for z Systems offered the developers a full-function mainframe development environment running on Windows without relying on the host mainframe platform. Meanwhile, Analyzer provided the knowledge and coding standards while Enterprise Developer offered the modern Eclipse-based IDE for developing and maintaining mainframe code.

Underpinning this, Open Source distributed source control systems, in concert with Jenkins, provided the continuous integration environment that enables on-demand building and testing.

The Result:

New hires found it easier to transition into mainframe projects, and experienced mainframe programmers were able to take advantage of modern, agile development techniques. This meant:

- Improved development productivity, using modern tools in an agile development environment
- Faster turnaround of maintenance changes and a significant improvement in quality of the delivered code back to the customer, minimizing the risk of production downtime
- Reduced mainframe dependency—a full application compile task has been reduced from day on the mainframe to 23 minutes under Enterprise Developer
- Elimination of unnecessary wait times and reduced consumption of development mainframe MIPS
- Skills is now no longer a concern—the average age of the mainframe developer for this client has been reduced to 26

How This Reflects the DevOps Model:

The outcomes in this scenario supported genuine IT objectives:

- **Shortening the Development Cycle**—exploiting workstation power to shorten mainframe development and unit test cycles, means that new products and features are made available faster
- **Increasing Application Release Velocity**—Increasing application development productivity through best-in-class developer tooling in the latest IDE technology improves output by as much as 40 percent.
- **Improving Defect Detection**—removing the bottlenecks reduces the risk to delivery schedules caused by quality, cost and time challenges
- **Decreased Development and Operations Cost**—incomplete integration equals inadequate developer adoption and increased productivity costs. A more modern, efficient IDE links the developer with trusted, working mainframe processes and technology and reduces costs.

Scenario C: Fast Provisioning and Test Environment Delivery

Providing continuous integration support and parallel projects testing.

Customer Profile:

This client is one of the largest general insurers in Australia. Their businesses in Asia and the UK underwrite \$9bn of insurance premiums each year and employ approx 13,600 people. Their core business production systems run on 4,000 MIPS mainframe systems.

The Challenge:

They experienced a three-fold increase in as many years for core insurance application business functionality. This demand outstripped the supply of mainframe test environments and with no significant investment in additional mainframe test capacity forthcoming, their ability to meet business requirements was negatively impacted. The result was slow time to market for application change. They needed a solution.

Resolution:

- Provide new test regions in hours rather than days or weeks, as was the case on the mainframe. With faster access to the test environment, developers could get quicker feedback on code changes.
- Offer the flexibility to be upscaled on demand, enabling concurrent testing of multiple projects, eliminating development bottlenecks
- Shift testing left towards the development teams, to enable earlier identification of integration issues

They chose Micro Focus Enterprise Test Server® (ETS). As an Enterprise Developer customer, they understood the value of modern development tooling and could see how the ETS Windows-based mainframe test capacity could remove their development and test bottlenecks.

The Result:

Now fully deployed, ETS is tightly integrated into their development processes. Application builds are triggered automatically by an application change, with components being auto-built under Enterprise Developer. A test region starts automatically in a virtualized environment with application load-modules from build process being copied across ready for testing.

Test regions are now provisioned in minutes giving development teams much faster access to an environment under which they can test their changes. This means issues are identified and fixed earlier in the process.

Using Enterprise Test Server in a virtualized environment offers access to an almost unlimited number of Test Server instances. They can be spun up on demand, enabling testing for multiple product releases in parallel.

The mainframe environment still provides the final test platform before changes are promoted to production but the flexibility that Enterprise Test Server provides and the testing they can do means that issues are now not found late in the cycle. The net result is that they have kept up with business demand for an increase in project delivery and have seen and increase in application quality.

How This Reflects the DevOps Model:

The outcomes in this scenario supported genuine IT objectives:

- **Increased Release Velocity**—Increasing application development productivity through best-in-class developer tooling in the latest IDE technology improves output by as much as 40 percent.
- **Improved Defect Detection**—removing the bottlenecks reduces the risk to delivery schedules caused by quality, cost and time challenges.
- **Reduced Deployment Failures and Rollbacks**—enabling mainframe IT organizations to perform pre-production testing on low cost commodity hardware improves success rates and avoids unnecessary cost and delay.
- **Decreased Development and Operations Cost**—incomplete integration equals inadequate developer adoption and increased productivity costs. A more modern, efficient IDE links the developer with trusted, working mainframe processes and technology and reduces costs.

In Conclusion

IT leaders looking to embrace the DevOps ethos of increased collaboration and unification are putting efficiency at the heart of their business strategy. By modernizing z/OS application service delivery they optimize every aspect of mainframe application delivery, drive down costs, remove bottlenecks and accelerate innovation.

For the mainframe owner, DevOps will never be a comprehensive silver bullet that fixes years of application complexity, siloed development processes and all the outdated practices that contribute to the problems plaguing the modern CIO, namely Regulatory Compliance, Time to Value and the loss of specialist skills to the industry. There is no plug-and-play DevOps program that will address all these problems completely.

11 www.ibm.com/ibm/devops/us/en/

12 www.microfocus.com/assets/visual-cobol-modernization-val_tcm6-174466.pdf

However, the software that will achieve the DevOps ideals of planning, collaborative development, testing, release and deployment, monitoring, feedback and optimization in a continuous cycle most certainly *is* available. As the scenarios in this paper have proved, enterprises with long-established mainframe-based practices and technologies can introduce contemporary levels of flexibility and agility.

Because while the names may change—DevOps is simply the latest label to be attached to the perpetual quest for cost savings and performance improvements—the drive for efficiency will always be with us. After all, when will we not want that? So it makes sense for organizations to look at the mainframe interpretations for DevOps, however it fits the profile of their organization.

Rosalind Radcliffe, IBM's Chief Architect for Collaborative Lifecycle Management and DevOps, recognizes that DevOps has a place on the mainframe landscape. For Radcliffe and IBM¹¹, DevOps offers “the capability for continuous software delivery that enables [mainframe owners] to seize market opportunities and reduce time to customer feedback.”

As Forrester principal analyst Amy DeMartine notes, DevOps offers something for many organizations. “DevOps doesn't seek to erase the differences between the two disciplines of software development and IT operations, but instead builds a bridge to make them work better together while continuing to follow traditional processes in each discipline independently.”

However, the same commentator observes that this is still a work in progress for many companies. “We are still getting there,” she said, “but I think in ten years' time we are going to look back and think, ‘Why did we do this any differently?’”

To discover how a DevOps regime could improve business outcomes for your organization, ask us about how our value profile service¹² could represent your first step on the journey to modernizing the application development process.

About Micro Focus

Since 1976, Micro Focus has helped more than 20,000 customers unlock the value of their business logic by creating enabling solutions that bridge the gap from well-established technologies to modern functionality. The two portfolios work to a single, clear vision—to deliver innovative products supported by exceptional customer service. www.microfocus.com



Micro Focus
UK Headquarters
United Kingdom
+44 (0) 1635 565200

U.S. Headquarters
Rockville, Maryland
301 838 5000
877 772 4450

Additional contact information and office locations:
www.microfocus.com