IBMSystems
m a g a z i n e

Making
DevOps Real
for System z

Linking development and operations for the continuous delivery of software innovation BY EVELYN HOOVER • PHOTOGRAPHY BY BRYAN REGAN

Business innovation is increasingly delivered through software. The rapid pace of market, economic and regulatory change is driving the need for business agility, pushing software delivery to become faster and more predictable. However, employees involved in various stages of software delivery frequently work in their own silos.

On one hand, developers are striving to deliver new application features as quickly as possible. On the other, the operations staff must ensure platforms and systems run smoothly and dependably. This relationship frequently resembles a baton handoff in a relay race.

DevOps changes that. It makes the relationship more like a rowing team pulling together to win a race. And IBM takes that concept one step further by extending it to the lifecycle of an application—from inception to consumption.

"DevOps is about accelerated, continuous delivery of software innovation," explains Hayden Lindsey, vice president and Distinguished Engineer, IBM Rational* Software.

IBM views DevOps as spanning the entire enterprise, accelerating delivery of multitier mobile and cloud solutions across distributed, Power* and mainframe platforms.

IBM Systems Magazine sat down with Lindsey to learn more about why it's a good fit for big iron.

IBM Systems Magazine (ISM): Can you describe DevOps and explain why there's so much buzz about it?
Hayden Lindsey (HL): All businesses want to be more responsive to client needs. That's nothing new. But some key market shifts underscore the need for increased business agility, and that relies on software. The exponential increase in the number of empowered users has driven higher expectations for a quality customer experience.

TAKEAWAY

➜ Rapid business innovation is necessary to respond to ever-changing markets, and software solutions can deliver that innovation.

➜ Many IT organizations silo their application development and operations staff, slowing the responsiveness of software delivery.

➜ Through a collaborative DevOps approach, organizations can better meet the need for agile and timely software delivery.

➜ IBM Rational Software solutions can be a key element to making DevOps synergy a reality to empower business innovation.

Hayden Lindsey,
VP and Distinguished
Engineer, IBM
Rational Software

> "DevOps is about establishing an enterprise capability for accelerated and continuous software delivery that enables our clients to seize market opportunities and reduce time to customer feedback."
>
> —Hayden Lindsey, VP and Distinguished Engineer, IBM Rational Software

And the rise of disruptive technologies like mobile, social, big data and cloud puts even more pressure on today's IT organizations. Delivering high-quality applications, especially customer-facing applications, is more critical than ever.

The conundrum is that IT has to do all of this while maintaining or even reducing budgets. Delivering applications every six, 12 or 18 months is no longer acceptable. From what I see, companies are having a hard time dealing with this, and the stakes are much higher today. Take mobile, for instance. The cost and risk of delivering poor-quality or unusable applications can have an immediate and negative viral response from a constantly connected global user community. Not having a mobile channel at all is even worse.

When you think about the agile movement, development shops are doing things faster, breaking work into smaller deliverables and then delivering that work as frequently as possible to

their clients. Meanwhile, operations teams—especially in larger companies—are working aggressively to ensure mission-critical systems never go down. Operations teams are naturally going to be pushing back on increasing the frequency of change to the production systems; thus, the dilemma.

DevOps is a set of principles to address this situation. It's about linking development and operations more seamlessly. It guides you to get applications into production more quickly by promoting collaboration between the development and operations teams. That's an industry view.

The IBM view extends this outward and across the enterprise—building feedback loops with customers and other stakeholders, from the lines of business to the help desk, to maximize the business value of software in an accelerated

and continuous manner (see "DevOps Lifecycle," below). Ultimately, DevOps is about establishing an enterprise capability for accelerated and continuous software delivery that enables our clients to seize market opportunities and reduce time to customer feedback.

ISM: DevOps is more prevalent in distributed systems environments. How does it resonate with System z* shops, which are more risk-adverse?

HL: The DevOps movement started in distributed shops. However, the principles and practices are relevant to a System z client, perhaps even more so. For example, enterprises that have their critical systems of record running on System z have the same challenges.

There's always a conflict in getting things done more quickly while maintaining uptime. One example is the integration of the entire end-to-end process, perhaps for a mobile device on the front end, known as a system of engagement, to a data server on the back end. And it's even more difficult and error-prone if it's done manually.

Further complicating things, IT organizations often deal with governance silos as well as cultural and procedural differences between teams. Again,

DevOps Lifecycle

Customers   Business Owners   Development/Test   Operations/ Production

◄ *Continuous Feedback and Improvements* ►

that presents a hindrance to doing things more quickly. For example, most mainframe shops keep separate LPARs available for production and development needs while managing them together. As development teams require changes to those LPARs, such as subsystem configuration or updated release levels, the operations teams have to prioritize that work behind keeping production running smoothly.

DevOps principles such as automation, integration, collaboration and analytics provide opportunities to break down some of the silos. This speeds delivery, maintains high quality and reduces the cost associated with development.

ISM: How can mainframe shops pick up the pace, so to speak?

HL: There are a few ways to get started. For example, let's talk about maintaining quality more effectively. You'd never want anything in production that hasn't been thoroughly tested. However, most mainframe shops rely on manual testing, which is error-prone and lengthy. Furthermore, it's certainly not the norm to trigger builds frequently and run tests in the background when the build is completed. Yet in the distributed arena this is a best practice.

One principle is to automate anything done manually or that could be error-prone. This leads to more fully embracing automated testing. To help with this, IBM offers tools, including Rational Test Workbench (RTW), to enable automation of functional tests, integration tests and so forth for CICS*, IMS*, DB2* and other applications as reached via interfaces such as Web services or CICS Transaction Gateway.

When you automate and then trigger test scripts to run whenever you get a valid build, you can build and test very frequently. This enables you to dramatically speed up delivery while actually increasing quality.

ISM: What you describe may cause some MIPS usage issues. Isn't that an inhibitor?

HL: If we didn't have an alternative, it would be an inhibitor. Automated testing and frequent builds are not the norm on the mainframe partly because

of the focus on development-related MIPS usage and how it's capped. Therefore, an alternative is needed to exploit the industry best practice of "test early, test often."

With the Rational Development and Testing Environment (RD&T),

instead of testing in a development LPAR on the mainframe, RD&T provides a dedicated z/OS* development and test environment for doing builds and tests. RD&T runs on an x86 Linux* machine and provides the stack of software that development needs: z/OS running IMS, DB2, etc. The best part is that it consumes no MIPS.

It's important to note this software is for development use only. It's not licensed, nor is it appropriate, for running a production workload. It's not a mainframe—you're running only commodity hardware, but you do get functional equivalence for development and testing purposes.

ISM: Operations still has to create and configure the test environments. Doesn't that slow down the agile development process?

HL: Maintaining test LPARs effectively is a big challenge. Changes to middleware versions or configurations can take weeks or longer. Again, development teams are dependent upon operations teams to make these changes quickly while operations teams are trying to make them reliable and prevent any impact to production systems.

There are a few strategies for dealing with this. One is to test application components in isolation with the use of "stubbed" or "mock" services. IBM offers Rational Test Virtualization Server (RTVS) to help with this. Used with RD&T, the combination provides two important benefits. It allows development teams to easily model and simulate behavior between dependent subsystems. It does this in isolation, without troubling the operations teams and without requiring access to those

other systems. This is accomplished via RTVS capabilities that allow you to simulate the response from an MQ message without really invoking MQ. This helps minimize subsystem dependency during testing and further reduces infrastructure costs and non-production MIPS consumption. It also decouples development teams from operations teams, allowing development teams to make changes rapidly and as they need.

This is an example where the System z platform is unique. We must enable decoupling of development and operations teams in the development and test stages of the lifecycle and then integrate development and operations at the appropriate events—such as for system, load and performance testing and, of course, deployment.

ISM: Wouldn't setting up environments like you've described

require a lot of attention from operations? Doesn't that introduce bottlenecks?

HL: That's a common question but it's a misconception. Yes, they are moving from operating a small number of development LPARs to operating potentially dozens of these RD&T instances. And that does spur concerns around management costs and so forth, but those objections come from the assumption that the environments will be managed the same way they are today—manually and one by one. That's not what we're suggesting.

IBM recently introduced IBM SmartCloud* Continuous Delivery (SCD). With SCD, we combine build management, cloud technology and automated quality testing in a single package to rapidly provision test environments in just minutes.

Development uses instances, maintained by operations, of a common system image. If an instance gets messed up, it's discarded and a new one is created from the common image. Instances aren't managed individually. These images can spread across 20 or 30 RD&T machines and can be provisioned in a matter of minutes. What's so amazing about this is it doesn't matter if there are three, 30 or 300 machines.

ISM: Would that also hold true for keeping the environments consistent and up to date?

HL: Exactly. It's challenging to ensure the development and target production environments are consistent. That's why system programmers will still be involved in the process of defining the test environment system images. These IBM tools will handle provisioning and distributing the operations-governed images to the development teams that need them.

ISM: Many other roles are also involved in software delivery on the mainframe. The people in those roles don't always communicate with one another. Are there ways to help them work better together?

HL: This is always challenging and more complex in a mainframe and multiplatform enterprise. As I noted earlier, the mainframe space has more cultural, organizational and process challenges to overcome than in a purely distributed shop. This is another area where IBM extends the scope of DevOps. IBM's Rational Collaborative Lifecycle Management Solution (CLM) supports truly agile development and this fits hand in glove with DevOps. CLM allows all key stakeholders in different roles to be brought together in the same collaborative environment to share data, provide feedback and review plans.

DevOps is ultimately about enabling software-driven business innovation and outcomes. Given this, you need to expand the view of each stakeholder, each person's role, and the potential impact of accelerated and continuous delivery. If you do, the results are obvious. For example, a major insurance client reduced its total time to validate each rate filing by 94 percent. And a top Latin American bank reduced—by 55 percent—the total time to develop, build and deploy software. Clearly these results have material impact to the business.

ISM: Where can readers find more information about DevOps?

HL: Information is available on IBM's website, however, I highly recommend readers attend Innovate 2013: The IBM Technical Summit. It's June 2 through 6 in Orlando, Fla., and it's a great place to learn more about DevOps. We feature dedicated System z and Power tracks targeted at developers and system programmers. There's a new track targeted at management, featuring DevOps, cloud, mobile, big data and social business solutions, as well as an Executive Summit for senior execs. You can get more details on specific topics and sessions online [ibm.com/software/rational/innovate/]. ℤ

Evelyn Hoover is executive editor of *IBM Systems Magazine.*