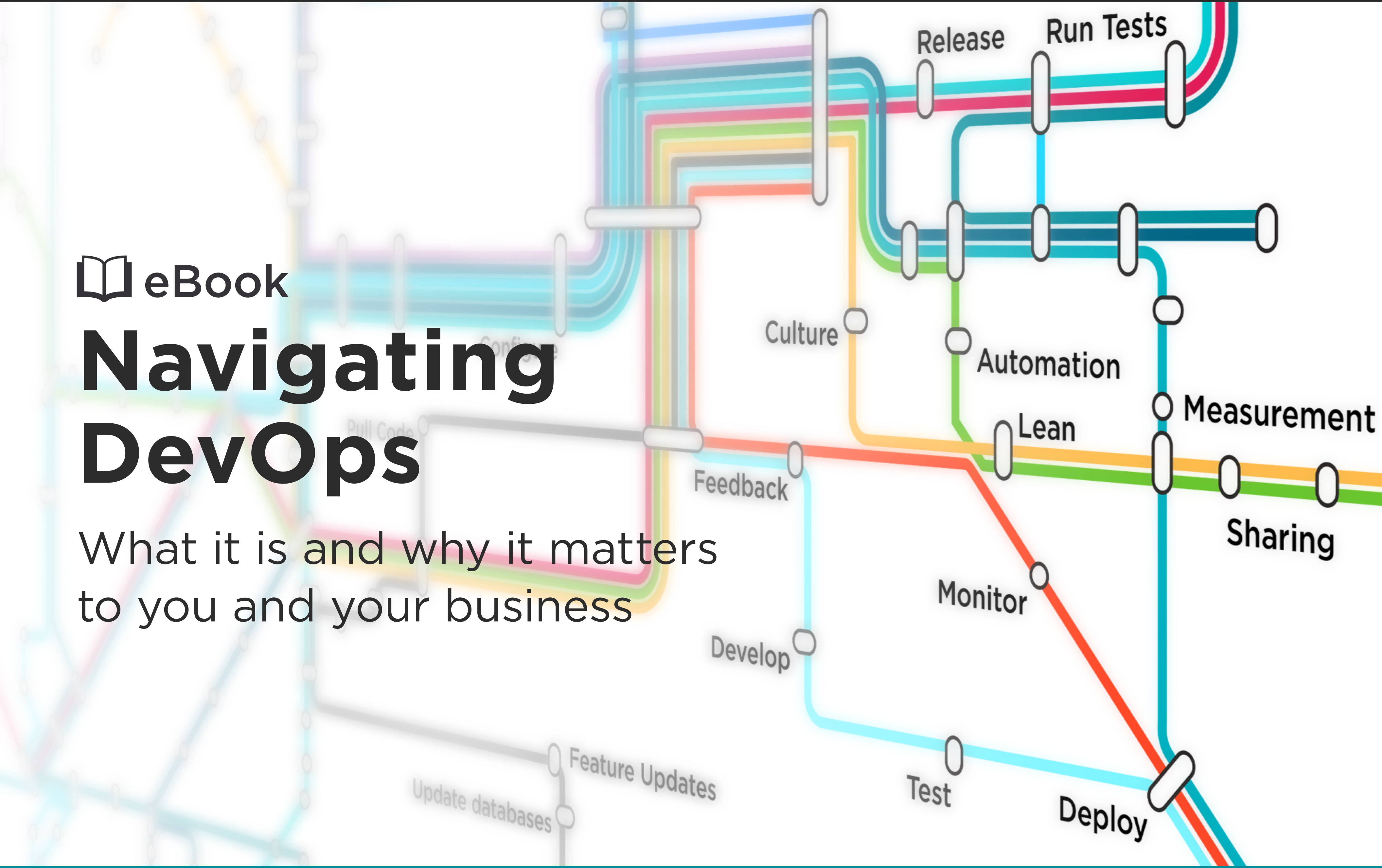
New Relic®



eBook

Navigating DevOps

What it is and why it matters
to you and your business

Table of Contents

INTRODUCTION		03
CHAPTER 1	What Is DevOps?	04
CHAPTER 2	Where Did DevOps Come From?	06
CHAPTER 3	What Problems Led to the Creation of DevOps?	08
CHAPTER 4	How Does DevOps "Work"?	10
CHAPTER 5	Who's Adopting DevOps?	14
CHAPTER 6	Why Are Your Peers Embracing DevOps?	16
CHAPTER 7	How Will I Benefit From DevOps?	20
CONCLUSION		22

Introduction

Even as DevOps adoption continues to accelerate in both large enterprises and web-native organizations, confusion lingers about what exactly the term means. Is DevOps a culture, a movement, an approach, a philosophy, or some amalgam of several of these things? Or does DevOps mean different things to different people?

However you define DevOps, achieving DevOps success undoubtedly requires a journey. And no matter where you are in your DevOps journey, this ebook is designed to help answer a number of fundamental questions, including:

- What is DevOps?
- Where did it come from?
- What problems led to DevOps?
- How does DevOps "work"?
- How widely used is DevOps today?
- Why are people adopting DevOps?
- What are the benefits?

CHAPTER 1

What Is DevOps?

What Is DevOps?

The word “DevOps” was coined in 2009 by Patrick Debois, who became one of its gurus. The term was formed by combining “development” and “operations,” which provides a starting point for understanding exactly what people typically mean when they say “DevOps.” Notably, DevOps isn’t a process or a technology or a standard. Many devotees refer to DevOps as a “culture”—a viewpoint that New Relic favors. We also use the term “DevOps movement” when talking about topics such as adoption rates and trends for the future and “DevOps environment” to refer to an IT organization that has adopted a DevOps culture.

This primer will have a great deal more to say about DevOps, but to get started, we need a serviceable definition. We like this one from Gartner:

> *“DevOps represents a change in IT culture, focusing on rapid IT service delivery through the adoption of agile, lean practices in the context of a system-oriented approach. DevOps emphasizes people (and culture), and seeks to improve collaboration between operations and development teams. DevOps implementations utilize technology—especially automation tools that can leverage an increasingly programmable and dynamic infrastructure from a life cycle perspective.”*[1]

[1] IT Glossary, Gartner, http://www.gartner.com/it-glossary/devops/

CHAPTER 2

Where Did DevOps Come From?

Where Did DevOps Come From?

Despite the mythical tone of some of the stories about its origins, DevOps was not created out of whole cloth. Rather, the seeds of DevOps were planted long ago and have been nurtured by forward-thinking IT experts in a number of disciplines. The two primary antecedents of DevOps are:

- **Enterprise systems management (ESM).** Many of the people involved in the initial definition of DevOps were system administrators. These operations experts brought key ESM best practices to DevOps, including configuration management, system monitoring, automated provisioning, and the toolchain approach.
- **Agile development.** One observer notes, "DevOps can be interpreted as an outgrowth of Agile—agile software development prescribes close collaboration of customers, product management, developers, and (sometimes) QA to fill in the gaps and rapidly iterate towards a better product ... [DevOps acknowledges that] service delivery and how the app and systems interact are a fundamental part of the value proposition to the client as well, and so the product team needs to include those concerns as a top-level item. From this perspective, DevOps is simply extending Agile principles beyond the boundaries of the code to the entire delivered service."[2]

Why does this matter? It goes to the basic credibility of the movement. Far from being some outlandish manifesto of a few IT geeks, DevOps is actually quite mainstream in its origins. DevOps unites an established IT operations discipline with a proven development methodology—each half of the acronym represents the best practices of its craft. And that union mirrors the fact that DevOps integrates development and operations into a single-minded entity with common goals: high-quality software, faster releases, and improved customer satisfaction—again, totally mainstream enterprise concerns.

> *"If you do Agile without DevOps, it's like you're trying to race with a tractor instead of a car. You can go and do the laps but it's not going to go very fast, you're probably going to consume a lot of fuel and it won't be a lot of fun."*
>
> —Frederic Veron, CIO, Fannie Mae[3]

[2] Ernest Mueller, "What is DevOps," The Agile Admin, https://theagileadmin.com/what-is-devops/, December 7, 2016.
[3] Clint Boulton, "True Agile Software Development Requires DevOps," CIO, http://www.cio.com/article/3141577/cio-role/true-agile-software-development-requires-devops.html, November 15, 2016.

CHAPTER 3

What Problems Led to the Creation of DevOps?

What Problems Led to the Creation of DevOps?

Developers and system administrators don't always see eye to eye on a lot of things, but they do agree that their customers on the business side of the house frequently pull them in different directions. On the one hand, business users demand change—new features, new services, new revenue streams—as fast as possible. At the same time, they also want a system that is stable and free from outages and interruptions. That creates a problem where companies feel like they have to choose between delivering changes quickly and dealing with an unstable production environment, or maintaining a stable, but stale environment.

Not surprisingly, neither choice is acceptable to enterprise executives. And more important, neither allows a business to provide the best solutions it can to its customers.

Developers are willing to push out software faster and faster—after all, that's what they are typically hired to accomplish. Operations, on the other hand, knows that rapid-fire changes without proper safeguards could destabilize the system, which goes directly against their charter.

DevOps was created to resolve this dilemma by integrating everyone associated with software development and deployment—business users, developers, test engineers, security engineers, system administrators, and sometimes others—into a single, highly automated workflow with a shared focus: Rapid delivery of high-quality software that meets all user requirements while maintaining the integrity and stability of the entire system.

How do these disparate groups join forces? By subscribing to a common set of principles that transcends traditional discipline boundaries and roles, for example:[4]

- Set expectations and priorities and the fundamental beliefs that guide them.
- Collaborate both within and between teams on problem solving.
- Automate common and repetitive processes to free up time for higher-level work.
- Integrate feedback into the work, measuring everything that is moved into production
- Share the data with everyone involved to foster a more effective culture of working well together across different skills and specialized knowledge.

[4] Aliza Earnshaw, "DevOps Principles for IT Managers," Puppet, https://puppet.com/blog/devops-principles-for-it-managers, February 25, 2016.

CHAPTER 4

How Does DevOps “Work”?

How Does DevOps "Work"?

Like all cultures, DevOps incorporates many variations on the theme. However, most observers would agree that the following capabilities are common to virtually all DevOps cultures: collaboration, automation, continuous integration, continuous delivery, continuous testing, continuous monitoring, and rapid remediation.

Collaboration

Instead of pointing fingers at each other, development and IT operations work together (no, really). While the disconnect between these two groups was the impetus for its creation, DevOps extends far beyond the IT organization, because the need for collaboration extends to everyone with a stake in the delivery of software (not just between dev and ops, but all teams, including test, product management, and executives):

> *"The foundation of DevOps success is how well teams and individuals collaborate across the enterprise to get things done more rapidly, efficiently and effectively."*[5]

Automation

DevOps relies heavily on automation—and that means you need tools. Tools you build. Tools you buy. Open source tools. Proprietary tools. And those tools are not just scattered around the lab willy-nilly: DevOps relies on toolchains to automate large parts of the end-to-end software development and deployment process.

Caveat: Because DevOps tools are so amazingly awesome, there's a tendency to see DevOps as just a collection of tools. While it's true that DevOps relies on tools, DevOps is much more than that.

Continuous Integration

You usually find continuous integration in DevOps cultures because DevOps emerged from Agile culture, and continuous integration is a fundamental tenet of the Agile approach:

> *"A cornerstone of DevOps is continuous integration (CI), a technique designed and named by Grady Booch that continually merges source code updates from all developers on a team into a shared mainline. This continual merging prevents a developer's local copy of a software project from drifting too far afield as new code is added by others, avoiding catastrophic merge conflicts."*[6]

The continuous integration principle of agile development has a cultural implication for the development group. Forcing developers to integrate their work with other developers' work frequently—at least daily—exposes integration issues and conflicts much earlier than is the case with water-

[5] Tony Bradley, "Scaling Collaboration in DevOps," DevOps.com, https://devops.com/scaling-collaboration-devops/, May 16, 2016.
[6] C. Aaron Cois, "Continuous Integration in DevOps," DevOps blog, Software Engineering Institute, Carnegie Mellon, https://insights.sei.cmu.edu/devops/2015/01/continuous-integration-in-devops-1.html, January 26, 2015.

fall development. However, to achieve this benefit, developers have to communicate with each other much more frequently—a process that runs counter to the image of the solitary genius coder working for weeks or months on a module before she is “ready” to send it out in the world. That seed of open, frequent communication blooms in DevOps.

Continuous Testing

The testing piece of DevOps is easy to overlook—until you get burned. As Gartner puts it, “Given the rising cost and impact of software failures, you can’t afford to unleash a release that could disrupt the existing user experience or introduce new features that expose the organization to new security, reliability, or compliance risks.”[7] While continuous integration and delivery get the lion’s share of the coverage, continuous testing is quietly finding its place as a critical piece of DevOps.

Continuous testing is not just a QA function; in fact, it starts in the development environment. The days are over when developers could simply throw the code over the wall to QA and say, “Have at it.” In a DevOps environment, quality is everyone’s job. Developers build quality into the code and provide test data sets. QA engineers configure automation test cases and the testing environment.

On the QA side, the big need is speed. After all, if the QA cycle takes days and weeks, you’re right back into a long, drawn out waterfall kind of schedule. Test engineers meet the challenge of quick turnaround not only by automating much of the test process but also redefining test methodologies:

> *“Continuous testing creates a central system of decision that helps you assess the business risk each application presents to your organization. Applied consistently, it guides development teams to meet business expectations and provides managers visibility to make informed trade-off decisions in order to optimize the business value of a release candidate.”*[8]

Although it may come as a surprise, the operations function has an important role to play in testing and QA. Operations can ensure that monitoring tools are in place and test environments are properly configured. They can participate in functional, load, stress, and leak tests and offer analysis based on their experience with similar applications running in production.

The payoff from continuous testing is well worth the effort. The test function in a DevOps environment helps developers to balance quality and speed. Using automated tools reduces the cost of testing and allows test engineers to leverage their time more effectively. Most important, continuous testing shortens test cycles by allowing integration testing earlier in the process.

Continuous testing also eliminates testing bottlenecks through virtualized, dependent services, and it simplifies the creation of virtualized test environments that can be easily deployed, shared, and updated as systems change. These capabilities reduce the cost of provisioning and maintaining test environments, and they shorten test cycle times by allowing integration testing earlier in the life cycle.

[7] “Evolving from Automated to Continuous Testing for DevOps,” Gartner and Parasoft, https://alm.parasoft.com/gartner-continuous-testing-devops-api-testing-service-virtualization, 2016.
[8] “Continuous Testing for IT Leaders,” Parasoft, https://alm.parasoft.com/continuoustestingbook, 2016.

Continuous Delivery

The team at Amazon Web Services defines continuous delivery as a DevOps "software development practice where code changes are automatically built, tested, and prepared for a release to production. It expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage. When continuous delivery is implemented properly, developers will always have a deployment-ready build artifact that has passed through a standardized test process."[9]

The actual release frequency can vary greatly depending on the company's legacy and goals. High-performing organizations using DevOps achieve multiple deployments per day compared to medium performers who release between once per week and once per month.[10]

Exactly what gets released varies as well. In some organizations, QA and operations triage potential releases: many go directly to users, some go back to development, and a few simply are not deployed at all. Other companies push everything that comes from developers out to users and count on real-time monitoring and rapid remediation to minimize the impact of the rare failure. And it's important to note that because each update is smaller, the chance of any one of them causing a failure is significantly reduced.

Continuous Monitoring

Given the sheer number of releases in a continuous delivery shop, there's no way to implement the kind of rigorous pre-release testing typically required in waterfall development approaches. In a DevOps environment, failures must be found and fixed in real time. How do you do that? A big part is continuous monitoring.

With continuous monitoring, teams measure the performance and availability of software to improve stability. Continuous monitoring helps identify root causes of issues quickly to proactively prevent outages and minimize user issues. Some monitoring experts even advocate that the definition of a service must include monitoring—they see it as integral to service delivery.

Like testing, monitoring starts in development. The same tools that monitor the production environment can be employed in development to spot performance problems before they hit production.

Two kinds of monitoring are required for DevOps: server monitoring and application performance monitoring. Monitoring discussions quickly get down to tools discussions, because there is no effective monitoring without the proper tools. For a list of DevOps tools (and more DevOps-related content), visit the New Relic DevOps Hub.

[9] "What is Continuous Delivery?" Amazon Web Services, https://aws.amazon.com/devops/continuous-delivery/.
[10] "2016 State of DevOps Report," Puppet Inc. and DORA, https://puppet.com/resources/whitepaper/2016-state-of-devops-report, June 2016.

CHAPTER 5

Who's Adopting DevOps?

Who's Adopting DevOps?

From early-stage startups to 100-year-old enterprises, DevOps is making significant inroads into IT organizations everywhere. One survey shows that 74% of companies have implemented DevOps in some fashion.[11]

What kinds of companies are embracing DevOps? While web-native "unicorns" like Etsy, Facebook, Amazon, and Netflix are oft-cited examples of DevOps leaders, today every type of business has gotten in the DevOps act. Mainstream media company Sony Pictures, financial services behemoth Barclays Bank, and building products manufacturer USG are some of the other DevOps success stories in the news.

Surprisingly, perhaps, enterprises are leading the charge with 81% reporting that they are adopting DevOps somewhere in their organization. Small and medium-sized businesses (SMBs) are also reaping the benefits of DevOps, with 70% saying they are using it.[12] Significantly, there's plenty of evidence showing that company size by itself is no predictor of DevOps success.

Even government and quasi-government organizations are embracing DevOps. Take Fannie Mae, for instance, which is using DevOps to transform its business and move from an organization that "does change very slowly to one that changes very quickly."[13]

> *"We now have the proof points to show that DevOps is truly not for just the unicorns of Google, Amazon, Facebook, and Netflix, but it really is for any technology organization, especially in large, complex organizations. What I'm so excited about is the fact that they're getting the same sort of outcomes that we've typically seen only in the unicorns."*[14]
>
> —Gene Kim, New Relic Modern Software Podcast, January 20, 2017

The U.S. Patent and Trade Office moved to DevOps and now sees an average of 1,000 automated builds per week.[15] Over at the General Services Administration (GSA), production containers, automated workflows, and microservices are just some of the ways that the government organization is modernizing its IT operations to deliver projects faster, at higher quality.[16]

But even as the rise of cloud and container technologies contribute to DevOps adoption worldwide, DevOps author and expert Gene Kim notes that DevOps still has plenty of room to grow. That often takes the form of DevOps driving new development efforts in organizations where it has already established a beachhead.[17]

[11] "RightScale 2016: State of the Cloud Report: DevOps Trends," RightScale, May 2016.
[12] Ibid.
[13] Phil Goldstein, "How Fannie Mae Used DevOps to Improve its IT Operations," FedTech, http://www.fedtechmagazine.com/article/2016/10/how-fannie-mae-used-devops-improve-its-it-operations, October 24, 2016.
[14] "Modern Software Podcast: Gene Kim on Measuring DevOps Success", New Relic Blog, https://blog.newrelic.com/2017/01/30/modern-software-podcast-episode-16-devops-gene-kim/.
[15] "Agile and DevOps at USPTO: Briefing for ACT-IAC Partners," June 23, 2016.
[16] Chris Riley, "Booz Allen Helps the GSA Put Your DevOps to Shame," DevOps.com, https://devops.com/booz-allen-helps-gsa-put-devops-shame/, May 13, 2016.
[17] New Relic Modern Software Podcast: Gene Kim on Measuring DevOps Success, https://blog.newrelic.com/2017/01/30/modern-software-podcast-episode-16-devops-gene-kim/.

CHAPTER 6

Why Are Your Peers Embracing DevOps?

Why Are Your Peers Embracing DevOps?

DevOps has something for everyone in the software chain: developers, operations, and testing. Furthermore, DevOps even touches the business side of the house: managers who monetize the software and executives who worry about the bottom line. Here are some of the benefits cited by each group.

Developers

Automated provisioning is a big win for programmers, because they can stand up a development environment themselves with no paperwork, no lengthy approval cycles, no waiting for IT to provision a server—no lost time. When developers can provision a working environment in minutes, with all the right resources—compute power, storage, network, applications—it changes the way that they work. They can be far more creative and innovative. It's much easier to try multiple options, run different scenarios, and test their code more thoroughly.

When developers first begin to work in a DevOps world, one real eye-opener for many is understanding just what goes on inside that black box labeled "Operations." That knowledge helps developers work effectively with operations in a joint problem-solving mode. Problems are resolved faster and cause fewer distractions. Best of all, the frantic late-night phone call from ops when a site goes down becomes a thing of the past—and that leads directly to greater job satisfaction and better quality of life for developers.

Operations

There's a widespread belief that system administrators constantly obsess about system stability—and, in fact, it's true. Their nightmare scenario is a software release that takes down the system within seconds of production deployment, developers who shrug off responsibility ("It's your code now!"), users in various degrees of outrage—and no clear path to a quick, effective resolution.

Early adopters of DevOps methods have found that the increased involvement by developers actually improves system stability. Critically, it turns out that smaller, more frequent releases introduce less variability into the system, lowering the risk of catastrophic failure. Even better, these more-limited releases can be made during the day, when everyone is working and available to solve problems, instead of in the middle of the night or on weekends.

Automation also helps by eliminating human errors common in manual operations, and has the added benefit of reducing the amount of time spent on routine tasks. There's a quality-of-life issue for system administrators as well, in the form of new skills, career opportunities, and a great deal more uninterrupted sleep and personal time. In a DevOps environment, operations rely on tools to a much greater extent than in traditional environment, often building their own and writing scripts that automate portions of the deployment process.

Put all this together and it should be no surprise that "Director of Operations" was the most common title for attendees at the recent DevOps Enterprise Summit.

Test Engineers

The impact that DevOps has had on the testing side of the house is enormous. As one commenter says:

> *"This is a world with several, almost continuous iterative releases, each within days, even minutes, of each other, all being pushed out to the final production environment into the demanding hands of paying customers. So many releases, so little testing time and so much pressure to deliver quality—has there ever been a more, theoretically, perfect case for automated testing?*[18]

DevOps requires new ways to test software, which challenges test engineers to innovate. With automated provisioning, test engineers can provision a test environment that is virtually identical to the production environment, resulting in more accurate testing and better ability to predict the performance of new releases. As with other groups, test engineer productivity increases thanks to automation and collaboration.

Product Managers

Technically, DevOps is just about the IT function of the enterprise. However, product managers, along with their marketing and business counterparts, can also see tremendous benefits:[19]

- **Faster feedback:** The moment a new product or feature gets delivered to customers, product managers start to get real feedback.
- **Increased responsiveness:** With continuous delivery, DevOps dramatically shortens the time-to-market for new features in response to customer needs.
- **Reduced waste and risk:** Development resources can fix problems or work on new features without waiting until the next big release.

Let's break that down a bit. In a DevOps environment, business stakeholders have greater influence on the development process. Thanks to the collaborative spirit of DevOps, developers actually care about business requirements and foster relationships with product managers. DevOps also gives product managers immediate feedback about the impact of new pricing, features, and product bundles, which allows them to test variations and gauge their effectiveness.

Line of business (LOB) managers love DevOps because software gets to market faster—giving them a competitive edge. Because DevOps improves system stability, customers experience fewer outages and are therefore more loyal—the perfect cure for high churn rates.

[18] ThinkSys Inc. , "Testing in the DevOps World—Test Automation, The Sequel, http://blogs.thinksys.com/testing-devops-world-test-automation-sequel/, January 23, 2017.
[19] Suzie Prince, "Why Continuous Delivery and DevOps are Product Managers' Best Friends," Mind the Product, http://www.mindtheproduct.com/2016/07/continuous-delivery-devops-product-managers-new-bff/, July 5, 2016.

Executives

When Patrick Debois and other IT wizards started the DevOps movement, they surely weren't concerned with how it would be received in the corporate boardroom. But DevOps is now a hot topic in those same boardrooms.

What do executives like about DevOps? For one thing, it helps the organization deliver high-quality products and get them to market much faster than competitors relying on traditional methods of software development—actions that impact the bottom line and build brand value. Another reason is the ability to attract and retain top talent: high-quality developers, system administrators, and test engineers want to work in the most modern, most productive fashion possible. Finally, when developers, operations, and QA work together, top executives rarely get pulled into inter-departmental disputes, leaving them more time to craft the focused business goals that everyone is now pulling together to reach successfully.

> *"Employees in high-performing (DevOps) teams were 2.2 times more likely to recommend their organization to a friend as a great place to work, and 1.8 times more likely to recommend their team to a friend as a great working environment. This is a significant finding, as research has shown "companies with highly engaged workers grew revenues two and a half times as much as those with low engagement levels."*[20]

[20] "2016 State of DevOps Report," Puppet Inc. and DORA, https://puppet.com/resources/whitepaper/2016-state-of-devops-report, June 2016.

CHAPTER 7

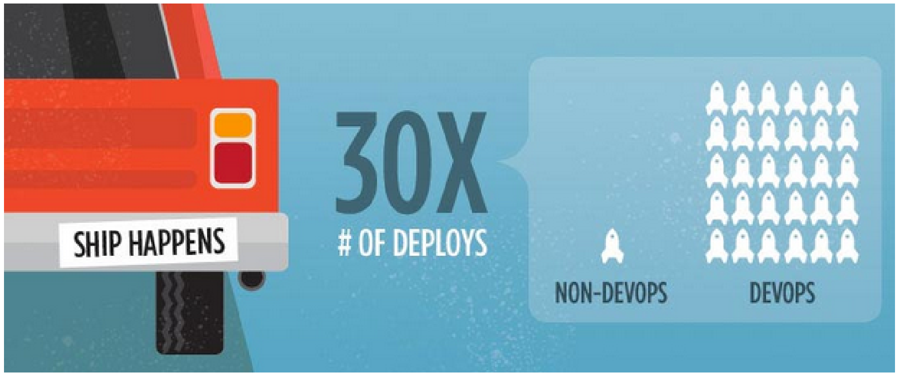
How Will I Benefit from DevOps?

How Will I Benefit from DevOps?

Credible sources report some pretty remarkable benefits achieved with DevOps. However, caution is in order. Suppose you overheard someone saying, "I'm getting 30 miles to the gallon." What car? What kind of driving? If he's talking about an F-150 truck driven off-road, the number is so high that you simply wouldn't believe it. On the other hand, for a new Prius driven exclusively on the highway, 30 mpg would be pretty disappointing. Context matters. So whenever you encounter claims about improvements related to DevOps, be aware that your results may vary.

That said, a survey by Puppet Labs[21] found that DevOps high performers release software 200 times more frequently than low performers, with 2,555 times faster lead times. The quality of the software products is higher, too, as shown by the finding that high performers have three times lower change failure rates than low performers. Finally, the net effect on system stability is highly positive: when the platform does go down, high performers restore service 24 times faster than low performers.

One thing is obvious: IT professionals who have adopted DevOps tend to be raving fans. It's not hard to see why, given the improvements cited in the same study:



Stability: High-performing organizations spend 22% less time on unplanned work and rework. As a result, they are able to spend 29% more time on new work, such as new features or code.

- **Security:** High performers spent 50% less time remediating security issues than low performers.
- **App deployment speed:** High performers deployed multiple times per day, on demand, compared to low performers who deployed between once per month and every 6 months.

[21] "2016 State of DevOps Report," Puppet Inc. and DORA, https://puppet.com/resources/whitepaper/2016-state-of-devops-report, June 2016.

Conclusion

Nearly a decade into the great DevOps experiment, the data is clear: DevOps is here to stay—and for some very good reasons. Many thought it impossible, but DevOps has succeeded in integrating business users, developers, test engineers, security engineers, and system administrators into a single workflow focused on meeting customer requirements. Why would they willingly do so? Because there's something in it for everyone. Developers and system administrators stop arguing and start supporting each other, lowering blood pressures all around. Business managers are happy because they get the software products that they need to sell products and services. Executives watch their beloved dashboard metrics—revenue, customer satisfaction, system reliability—heading steadily north. And everyone is able to deliver the best results and overall experience possible to the customer.

> *"DevOps is not a goal, but a never-ending process of continual improvement."*[22]
>
> —Jez Humble, Founder and CTO, DevOps Research and Assessment

Gains like these, however, don't come easily. To successfully deploy code more frequently while keeping your systems humming, you need the ability to accurately monitor all the changes going on in your environment. The New Relic Digital Intelligence Platform gives developers and operations full-stack visibility—from the digital customer experience to the applications and dynamic infrastructure, through integrated alerts and dashboards—which can help everyone within an organization enjoy a shared understanding of how software is deployed and performs in real time. To learn more about DevOps and how New Relic can help your organization successfully make the transition, visit http://newrelic.com/devops.

[22] "10 Deep DevOps Thoughts From Jez Humble," New Relic blog, https://blog.newrelic.com/2015/04/28/devops-jez-humble/.

About New Relic

New Relic is a leading digital intelligence company, delivering full-stack visibility and analytics with more than 14,000 paid business accounts. The New Relic Digital Intelligence Platform provides actionable insights to drive digital business results. Companies of all sizes trust New Relic to monitor application and infrastructure performance so they can quickly resolve issues, and improve digital customer experiences. Learn more at newrelic.com.

New Relic®