

# DEVOPS

## CONTINUOUS DELIVERY

eGuide



TECHWELL™

3

It's a Pivotal Time for DevOps

5

6 Steps to a Successful DevOps Adoption

8

Engineering Architecture Systems for a Faster Build

9

The Essential Role of QA in Digital Transformation

11

Build the Right Things and Build Them Fast: Accelerate the Continuous Delivery Pipeline

12

Finding the Bottlenecks in the Agile and DevOps Delivery Cycle

13

Insight from around the Industry

14

Additional DevOps Resources

The three years since we published our first DevOps eGuide have been a time of evolution—and revolution—for the DevOps movement. No longer just an industry buzzword, DevOps practices are being embraced by start-ups and enterprise-level organizations alike. What agile ignited with its focus on frequent releases, DevOps has turned into a full-fledged push for continuous everything—from build to delivery.

This eGuide provides resources to help teams move from their initial DevOps adoption through a full digital transformation. For organizations with established DevOps processes, this guide also offers tips to speed up the delivery cycle by identifying bottlenecks and building quality in from the start.

## In this DevOps eGuide

### It's a Pivotal Time for DevOps

Customers expect real-time software updates. As DevOps becomes the engine for delivering business value, continuous innovation is needed. And this has to begin at the start of every project.

### 6 Steps to a Successful DevOps Adoption

Implementing DevOps practices can significantly accelerate software releases while still assuring applications meet quality objectives. But DevOps can't be bought, bolted on, or just declared. If you're considering a move to a DevOps delivery model, here are six approaches for ensuring a successful DevOps adoption within an organization.

### Engineering Architecture Systems for a Faster Build

In the era of continuous integration and continuous deployment, big applications are creating bloated build pipelines. The problem is when code becomes so entangled that every change impacts large portions of the system, meaning there's a lot to rebuild. If you reshape the code architecture, you can reduce build times.

### The Essential Role of QA in Digital Transformation

Many organizations are undertaking a total digital transformation in order to accelerate processes, innovations, and opportunities. But if you think you can update your development efforts without making a change to your QA program as well, your change will not be successful. QA is essential to reliability.

### Build the Right Things and Build Them Fast: Accelerate the Continuous Delivery Pipeline

When most people think about continuous delivery, they think of improving the build-test-deploy-operate cycle. They don't think about how to improve the intake process. Ensuring that quality is built into the application—not tested for after the fact—is the key to achieving accelerated continuous delivery.

### Finding the Bottlenecks in the Agile and DevOps Delivery Cycle

To achieve incremental software development and continuous feedback, you need to eliminate the tasks that create bottlenecks, which hinder the flow of development. A chain is no stronger than its weakest link, and identifying these "weak links" is a critical step toward achieving agility and increasing efficiency.

### Insight from around the Industry

What industry insiders have to say about DevOps and today's "continuous" culture.

### Additional DevOps Resources

# It's a Pivotal Time for DevOps

By Eric Robertson



The complex nature of software development and delivery, especially at an enterprise scale, has resulted in DevOps gaining importance in recent years. How many of us have been frustrated by online banking or some other vendor interaction? How many of us have left a vendor for another that offers a better customer experience? Today's digital world shows us how the speed and quality of software delivery can either help or harm customer satisfaction and affect business outcomes.

Better customer experiences are driven by better software, and Microsoft CEO Satya Nadella saw it coming at the company's annual Convergence conference in 2015 when he stated, "Every business will become a software business, build applications, use advanced analytics, and provide SaaS services." [1]

All organizations are impacted by software, and all businesses are in the software business.

The quality and functionality of a company's software affects everything from competitive differentiation to customer support and, ultimately, employee satisfaction. So why aren't all private and government organizations delivering better offerings and better service at greater speeds?

## The Importance of DevOps

Traditional efforts to deliver innovative software solutions are often hampered by the limitations of the disparate tools, methods, and platforms in use today. Teams tend to be spread out geographically, and today's software development requires collaboration between R&D and IT operations.

DevOps will become an even higher priority to the enterprise as IT professionals learn how it helps bring innovative ideas to life by accelerating and improving software development.

Companies that expand their DevOps practices will experience the benefits of better teamwork between development and other groups across the enterprise.

Next-generation DevOps tools are starting to deliver comprehensive views of software release cycles. They combine those views with operational data that teams can use to make better-informed decisions. Key performance indicator (KPI) data will come into play, providing a link between an organization's software development lifecycle and its business. Fundamentally, DevOps is changing to usher in these and other advances that connect software development to the heart of the enterprise.

Here are some of the trends I see developing in the next few years as the DevOps market evolves.

3

It's a Pivotal Time for DevOps

5

6 Steps to a Successful DevOps Adoption

8

Engineering Architecture Systems for a Faster Build

9

The Essential Role of QA in Digital Transformation

11

Build the Right Things and Build Them Fast: Accelerate the Continuous Delivery Pipeline

12

Finding the Bottlenecks in the Agile and DevOps Delivery Cycle

13

Insight from around the Industry

14

Additional DevOps Resources

3

It's a Pivotal Time for DevOps

5

6 Steps to a Successful DevOps Adoption

8

Engineering Architecture Systems for a Faster Build

9

The Essential Role of QA in Digital Transformation

11

Build the Right Things and Build Them Fast: Accelerate the Continuous Delivery Pipeline

12

Finding the Bottlenecks in the Agile and DevOps Delivery Cycle

13

Insight from around the Industry

14

Additional DevOps Resources

## The Left Shift

Companies are beginning to leverage specific DevOps tools, which has led to a rapid uptick in automated testing and continuous delivery. The adoption of automated testing created a leftward shift in the pipeline, resulting in smaller-scale tests that are completed earlier and faster. Quality continues to be a focus, and micro-services are helping accelerate that drive by enabling the deployment of higher-quality code at a smaller risk to the business.

*Next-generation DevOps tools are starting to deliver comprehensive views of software release cycles. They combine those views with operational data that teams can use to make better-informed decisions.*

These shifts in testing and code deployment have pushed feedback further left in the pipeline, so teams are receiving responses earlier in the delivery lifecycle. As more organizations pick up on this shift, software quality will improve, and risks will be reduced further down the pipeline. This lowers the potential impact to the customer in terms of outages and operational costs by lessening the number of service requests.

## Driving DevOps with Analytics

Focusing on analytics provides a more holistic and comprehensive approach to DevOps. Over the next year, we will see more connected tools and processes, as well as KPI data that can enable new levels of decision making by leveraging operational data to provide intelligent correlation and traceability. Through KPI data, for example, organizations can unwrap the hidden issues within a software release that led to a jump in service tickets. With this type of powerful analytic data, detailed and revealing reports can be used to collect metrics from the tools and activities from chained associations.

## DevOps Becomes an Integral Part of the Project Lifecycle

While DevOps tools are meeting the needs of organizations involved with the software development lifecycle, many organizations have evolved. They now need to understand their DevOps value stream across the software development and delivery lifecycle—from planning to operations. This enables organizations to deliver end-to-end traceability across every DevOps tool chain component and to

leverage objective metrics and KPIs. This ensures that the delivered value is always operational and meets service-level agreements for the business. Essentially, this means continuous monitoring and feedback across DevOps tool chains.

This year, DevOps excellence is expected and has become the catalyst for successful software solutions. We are witnessing the importance of DevOps even at the executive level of enterprise software solutions. Consider bringing DevOps to the very beginning planning stages of your project lifecycle and coupling KPIs with analytics to measure operational success. By prioritizing these considerations, enterprise leaders will better leverage existing investments and set themselves up for future success in an industry that is constantly changing.

## References

1. Saran, Cliff. "Satya Nadella: Every business will be a software business." Computer Weekly . March 18, 2015. <http://www.computerweekly.com/news/2240242478/Satya-Nadella-Every-business-will-be-a-software-business>.

# 6 Steps to a Successful DevOps Adoption

By Alan Crouch

Figuring out the most optimal way to enable agility and rapidly deliver services to customers—without compromising quality—continues to be one of our industry's biggest challenges. Many IT leaders agree that implementing DevOps practices can significantly accelerate software releases while still assuring our applications meet quality objectives.

If you're considering a move to a DevOps delivery model, here are six approaches I've found to be critical for ensuring a successful DevOps adoption within an organization.



## 1. Embrace a DevOps Mindset

DevOps doesn't begin by just stating, "Let's do DevOps," and jumping in with tools. Your entire organization needs to have a clear understanding of what DevOps is and what specific business needs it can address, and, most importantly, everyone needs to be willing to change the way things have always been done.

One way to begin this process is to identify your current application *value streams*—the series of activities necessary for moving your products from development all the way into production. Understanding where in this delivery process there are constraints, bottlenecks, and wait queues will allow you to which activities you should concentrate on improving.

Identifying areas where your current delivery process is inefficient and must be improved is your opportunity to truly make change in your organization. But in order to do so, you must be willing to experiment. Short-term failure is acceptable, as long as you're learning from it and improving.

Instead of accepting inefficient ways of doing business because that's the way it's always been done, you will need to encourage your organization to ask questions like: "Why do we do this [process]? What's its business value? How can we make it more efficient?"

Organizations often equate DevOps with automation. While automation can help accelerate manual processes, DevOps is fundamentally about collaboration and communication. If you don't embrace strong communication and collaborative practices among everyone in the software development, testing, delivery, and operational process, automating your processes will not yield the business benefits you desire.

## 2. Make the Most of Metrics

One of the most overlooked initiatives in DevOps adoption is selecting the right metrics to record and track progress. Establishing

3

It's a Pivotal Time for DevOps

5

6 Steps to a Successful DevOps Adoption

8

Engineering Architecture Systems for a Faster Build

9

The Essential Role of QA in Digital Transformation

11

Build the Right Things and Build Them Fast: Accelerate the Continuous Delivery Pipeline

12

Finding the Bottlenecks in the Agile and DevOps Delivery Cycle

13

Insight from around the Industry

14

Additional DevOps Resources

the right baseline DevOps metrics early on and not being afraid to measure things that might initially not make you look very good is the key to being able to show demonstrative progress over time and real business benefits to senior leadership.

From my experience, these are some of most useful DevOps metrics:

- Production failure rate: how often the software fails in production during a fixed period of time
- Mean time to recover: how long it takes an application in production to recover from a failure
- Average lead time: how long it takes for a new requirement to be built, tested, delivered, and deployed into production
- Deployment speed: how fast you can deploy a new version of an application to a particular environment (integration, test, staging, preproduction, or production environments)
- Deployment frequency: how often you deploy new release candidates to test, staging, preproduction, and production environments
- Mean time to production: how long it takes when new code is committed into a code repository for it to be deployed to production

There are many other metrics you can collect too, but avoid collecting vanity metrics that look impressive but don't tie to business benefit, as well as metrics that are easily gamed—meaning they make your team look good but don't actually contribute to business improvements, such as the number of commits your team makes.

Once you've determined the metrics you wish to collect and have a baseline of where you currently stand, set goals for each metric so the team knows what to strive for.

Most importantly, constantly share your DevOps goals, metrics, and progress with everyone involved. Set up metrics dashboards that display current metrics and progress toward your goals. Providing complete transparency is sometimes a difficult thing for teams to

do, but it will foster more effective communication and collaboration, breaking down barriers between the dev and ops teams in the process.

### 3. Understand and Address Your Unique Needs

Contrary to what those selling DevOps products will tell you, there is no "one size fits all" solution for DevOps. You cannot just drop in an automated tool or hire a self-proclaimed "DevOps engineer" and expect success. Every organization will have a unique DevOps journey tied to its specific business and culture, and that journey will be far more about changing people's habits and communication patterns than what tools help you enable automation.

DevOps is a way of accelerating the creation and delivery of quality software, but it only succeeds if you focus on what makes business sense for your organization. For instance, if your customers can't consume ten to twenty updates to your product a day, don't make doing so your goal! Instead, focus on improving the usability, security, or some other key attribute your customer cares more about.

### 4. Adopt Iteratively

When getting started, don't try to boil the ocean with a complete, enterprise-wide DevOps initiative. Identify a pilot application, form a cross-functional DevOps team that includes dev, test, and operations, examine your value stream to determine constraints and bottlenecks, and create an initial deployment pipeline that addresses some of your process constraints. Measure progress and success, wash, rinse, and repeat.

Generally, you should look at tackling your biggest value stream constraints first, as doing so will have the largest business impact. Some of these constraints will be easy to resolve, while others will take a significant amount of time to correct—and often a whole lot of convincing others to change.

You'll want to go through a few iterations to build confidence in the framework and the pilot before you start expanding to other projects.

3

It's a Pivotal Time for DevOps

5

6 Steps to a Successful DevOps Adoption

8

Engineering Architecture Systems for a Faster Build

9

The Essential Role of QA in Digital Transformation

11

Build the Right Things and Build Them Fast: Accelerate the Continuous Delivery Pipeline

12

Finding the Bottlenecks in the Agile and DevOps Delivery Cycle

13

Insight from around the Industry

14

Additional DevOps Resources

Ensure you're making progress against your metrics before moving on and applying those lessons to other teams. Most importantly, make sure those involved are influencers who can take the principles back to their respective teams; keeping all your expertise locked up on your pilot won't help you expand to the enterprise effectively.

*Automation is the cornerstone of accelerating your delivery processes, and everything—infrastructure, environment, configuration, platform, build, test, process, etc.—should be defined and written in code.*

If you're beginning your DevOps journey from inside a software development organization, consider starting from the beginning of your delivery process and moving toward production. Properly implementing branch management strategies and build automation is key to fast feedback that will enable efficient downstream processes in the future.

Once sound build automation is in place, a more comprehensive continuous integration capability that includes continuous testing is in order. Getting your continuous integration process working effectively will allow you to begin shifting left your assurance activities over time and speeding up delivery.

### 5. Emphasize Quality Assurance Early

Based on my observations in most organizations, testers get the least amount of time to do quality assurance, and eventually, the quality of the product suffers. Organizations that struggle with DevOps often focus their efforts on automating deployments, overlooking the needs of QA.

While it is impossible to automate all your testing in DevOps, it is critical to automate all tests run as part of your continuous integration process (unit tests, static code analysis, etc.), as well as regres-

sion testing and smoke testing performed on each environment within your delivery process. Automating at least some functional testing and nonfunctional tests associated with security, performance, and other quality characteristics can often be achieved to speed up these activities.

Run lengthy, long-running tests that may require large, production-like environments late in your DevOps process so as to not slow down earlier feedback cycles that must be quick.

### 6. Take a Smart Approach to Automation

Automation is the cornerstone of accelerating your delivery processes, and everything—infrastructure, environment, configuration, platform, build, test, process, etc.—should be defined and written in code. If something is time-intensive, broken, or prone to error, start automating there first. This will quickly benefit your team by reducing delivery times, increasing repeatability, and eliminating configuration drift.

Standardize your approach to automation in order to ensure dev, ops, QA, and everyone in between has a common frame of reference and a common language. It's important that you use software engineering best practices when building DevOps automation. Infrastructure as code should be designed and implemented with coding standards, effectively tested, under configuration control, and well documented. The quality of your automation should be just as important as the quality of your application.

### Start toward Your Business Goals

DevOps can't be bought, bolted on, or just declared. It's something you can only achieve with a lot of hard work and discipline. It will take time and there will be some challenges, but if you follow the approaches I've laid out above, you will cover your bases and will be successful. Remember, DevOps is a journey that is never done, as there are always ways to improve what you do. But if you never get started, you won't reach your business goals.

# Engineering Architecture Systems for a Faster Build

by Abraham Marin-Perez

Speed is one of the key factors in any successful endeavour. But we tend to see the importance of speed and performance only when running applications, often forgetting that it is just as important at the time of building them.

In the era of continuous integration and continuous deployment, big applications are creating bloated build pipelines. This means feedback gets to developers late, affecting the ability of a business to react to events.

Companies are beginning to realize this threat and are acting on it. Unfortunately, many of the approaches we're seeing are rather idealistic concepts that sound good on paper but don't usually deliver the expected value.

One example is teams rewriting their entire stack into a microservices architecture, thinking that smaller components will be faster to build. However, they fail to realize that once the system grows big enough, those microservices will have shared components and interdependencies that will slow the build down.

Another instance is teams switching to experimental build tools or breaking code encapsulation with monolithic repositories. While these do provide some benefits, as with any tool, they come with disadvantages and the trade-off may not always be worth it.

There is a simpler way to keep things fast, if we can just understand the root of the problem. The whole idea of a continuous integration pipeline is that, whenever a change is made, everything

impacted by that change is rebuilt in order to ensure that we are always up to date.

It follows that the real problem is where code becomes so entangled that every single change impacts large portions of the system, meaning there's a lot to rebuild.

The solution is therefore simple (in principle!): just reshape the architecture of your code so that code changes affect a smaller portion of the overall system. In turn, only a smaller portion needs to be rebuilt, resulting in shorter build times.

For instance, if you have a library that is used by several other components, every time you modify that library you'll have to rebuild all the dependent components. If, however, you separate that library into its API and its implementation, then you might be able to reduce its impact—when you make a change to the implementation and it doesn't affect the overall behavior, dependents won't be affected, so you won't need to rebuild them.

Obviously, IT is a very fast-paced industry. New ideas and technologies come up every day, and we need to evaluate them in order to keep up. However, the fact that new things appear doesn't necessarily mean we always need to drop everything that came before it.

In the end, the oldest trick in the book is still one of the most effective: If you want to run things smoothly, make sure everything is appropriately tidied up as you go. Not only will you save a lot of pain, but it's the only way to keep that all-important performance up.

3

It's a Pivotal Time for DevOps

5

6 Steps to a Successful DevOps Adoption

8

Engineering Architecture Systems for a Faster Build

9

The Essential Role of QA in Digital Transformation

11

Build the Right Things and Build Them Fast: Accelerate the Continuous Delivery Pipeline

12

Finding the Bottlenecks in the Agile and DevOps Delivery Cycle

13

Insight from around the Industry

14

Additional DevOps Resources



# The Essential Role of QA in Digital Transformation

By Sunil Sehgal

With technology being embraced in all aspects of our lives, its influence is expanding. Organizations want to fully take advantage of this growth, leading many to go through a total digital transformation in order to accelerate processes, develop innovations, and embrace opportunities.

Quality assurance has an integral role in the journey of digital transformation. Implementing quality checkpoints along the path assures the desired business outcomes, as well as end-user satisfaction—the ultimate objective.

A complete digital transformation affects business processes at every level. Quality assurance efforts strengthen the product at each transition throughout the development lifecycle and mitigate challenges posed by new technology. QA is in the best supporting role in any organization's digital initiative to safeguard performance and security.

Luckily, enterprises advancing in the direction of expanded digitalization already recognize the associated risks and see the big picture of quality challenges. Therefore, many of them do not hesitate to expand their quality measures and spend a larger chunk of their IT budget on QA alone.

Good quality assurance measures meet the three major objectives of a company's digital transformation: protecting the corporate image, increasing quality awareness, and ensuring end-user satisfaction. Let's break down these critical aspects.



3

It's a Pivotal Time for DevOps

5

6 Steps to a Successful DevOps Adoption

8

Engineering Architecture Systems for a Faster Build

9

The Essential Role of QA in Digital Transformation

11

Build the Right Things and Build Them Fast: Accelerate the Continuous Delivery Pipeline

12

Finding the Bottlenecks in the Agile and DevOps Delivery Cycle

13

Insight from around the Industry

14

Additional DevOps Resources

3

It's a Pivotal Time for DevOps

5

6 Steps to a Successful DevOps Adoption

8

Engineering Architecture Systems for a Faster Build

9

The Essential Role of QA in Digital Transformation

11

Build the Right Things and Build Them Fast: Accelerate the Continuous Delivery Pipeline

12

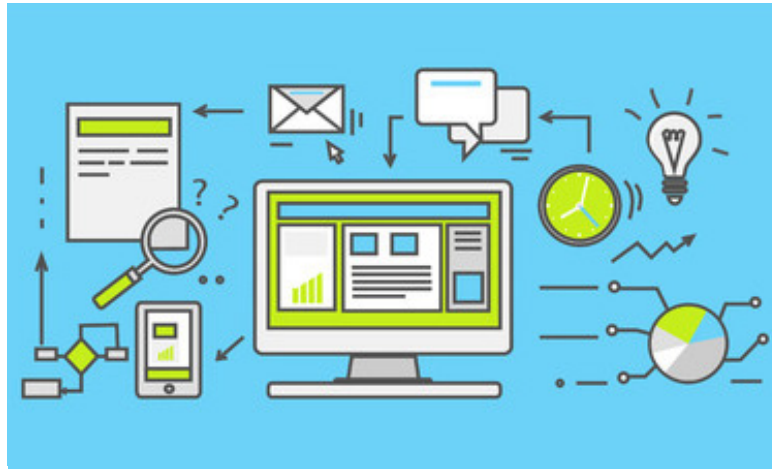
Finding the Bottlenecks in the Agile and DevOps Delivery Cycle

13

Insight from around the Industry

14

Additional DevOps Resources



**Protecting the customer experience and brand:** QA efforts measure end-user experiences across multiple digital platforms. This improves accessibility, reduces response time, and increases the consistency of brand. A good QA program can address the challenges for testing customer experience in multi-channel solutions:

- Getting the right coverage of end-user expectations and requirements in the test set
- Implementing and using test tools for customer experience testing
- Designing test cases
- Establishing environments and test data for customer experience testing

*A complete digital transformation affects business processes at every level. Quality assurance efforts strengthen the product at each transition throughout the development lifecycle and mitigate challenges posed by new technology.*

**Lifecycle automation:** QA speeds up the transformation process not only by application-level automation, but also by lifecycle automation. Apart from taking advantage of conventional automation, QA focuses on testing the entire lifecycle, thus ensuring cross-platform compatibility and customer-experience testing.

Quality user experience is a competency that's not just related to the dev or testing team. Rather, it's something that you need to work for as an entire unit. This is exactly what can be achieved with DevOps and test-driven development.

**Advanced analytics data support:** QA makes use of advanced analytical technologies to mine data from various social and commercial platforms that can be used in productive decision-making. This gives quality insight into all stages of the end-to-end customer experience.

A good QA program provides the most complete quality view, allowing dev teams to execute and analyze manual, automated, and performance tests for digital channels side by side, on desktop browsers and real mobile devices, under real end-user conditions. With this expansion, users can apply a single quality strategy to deliver seamless, responsive web and multichannel experiences.

QA helps test teams keep pace with development by assuring reliability of the product and elevating the digital customer experience. With this coverage, organizations can confidently venture into a digital transformation.

# Build the Right Things and Build Them Fast: Accelerate the Continuous Delivery Pipeline

by Alex Martins

When most people think about continuous delivery, they think of improving the build-test-deploy-operate cycle. They don't think about how to accelerate and improve the intake process. There is not a lot of focus on whether teams are using the right inputs (i.e., clear, unambiguous requirements) before they start coding, testing, and deploying.

Ensuring that quality is built into the application—not tested for after the fact—is the key to achieving the desired acceleration in continuous delivery (CD).

Testing, quality assurance, and quality engineering organizations are trying to transform themselves in order to enable acceleration in the CD pipeline while continuing to build quality into the application. Actually, these organizations are in a prime position to drive changes across their employee culture, process, and technology because their resources are inherent across the entire software development lifecycle. Organizations that have already realized this have put in place programs to upskill the testing staff.

Organizations working to achieve CD quickly know they are great at building things right and with speed, but they are still unsure about whether they are building the right things. There is a difference between the two, and in my experience, this is a gap in CD initiatives.

So, how do you ensure you are building the right things and you're building them fast? You focus on improving and accelerating the

requirements-gathering process, regardless of whether you're in a traditional or agile organization.

A good example is the way requirements are communicated across different teams. Requirements are the foundation of everything in the software development lifecycle, yet after thirty years, most are still being communicated through writing. They are written in Word or Excel documents or in requirements management tools—all completely manual processes.

Naturally, a manual process becomes a bottleneck in a highly automated CD pipeline where the ultimate goal is speed with quality. Not only that, but requirements written in text format many times are unclear and open to interpretation. Ambiguity is the cause of many defects introduced in the application code.

Instead of trying to run more tests in order to improve quality by finding defects—a *reactive* solution—what if we ran fewer tests because we have built things correctly from the beginning, therefore being *proactive* and preventing the high amount of defects from getting into the code?

We'd be achieving maximum acceleration of the continuous delivery pipeline by reducing waste caused by defect troubleshooting and rework and ensuring much higher quality. Software engineering practices are mature enough today to deliver on this old promise!

3

It's a Pivotal Time for DevOps

5

6 Steps to a Successful DevOps Adoption

8

Engineering Architecture Systems for a Faster Build

9

The Essential Role of QA in Digital Transformation

11

Build the Right Things and Build Them Fast: Accelerate the Continuous Delivery Pipeline

12

Finding the Bottlenecks in the Agile and DevOps Delivery Cycle

13

Insight from around the Industry

14

Additional DevOps Resources

# Finding the Bottlenecks in the Agile and DevOps Delivery Cycle

by Tanya Kravtsov

The iterative, collaborative models of DevOps and agile development changed the way we deliver software. But adopting these models requires more than just continuous integration, continuous testing, and continuous delivery—it requires a complete mind shift.

In order to achieve incremental software development and continuous feedback, you need to eliminate the tasks that create waste or bottlenecks, which hinder the flow of development. A chain is no stronger than its weakest link, and identifying these “weak links” is a critical step toward achieving agility and increasing efficiency.

The first step is gaining a full understanding of the system, including all the players and the handoffs between them. Mapping the delivery pipeline in the form of a flow chart should quickly highlight the points of improvement.

Talking to people from different groups within the organization who play a role in the system also can go a long way toward understanding where the bottlenecks are. Interviewing members of the development, quality assurance, product, IT operations, marketing, and senior leadership teams will identify the common themes and present ideas for improvement.

Brainstorming techniques such as mind maps, storyboarding, and innovation games such as Speed Boat can help facilitate these discussions. Retrospectives, an essential part of agile Scrum practices, are another great tool that encourages teams to provide their feedback on what works and what doesn't.

The most common bottlenecks in the software delivery lifecycle often turn out to be environments, testing, and communication. (How many

times have we heard “But it works on my machine”?)

To avoid such issues, it's important to automate the environment set-up process by identifying all the steps necessary to set up a machine and scripting them using infrastructure as code and configuration automation tools, in conjunction with service virtualization. It's also essential to either give developers and testers the ability to spin up preconfigured environments on demand or to have dedicated, stable environments with consistent monitoring and alerts set up in case something doesn't work as expected.

Testing is often considered a bottleneck due to its manual nature, and this is compounded when testing starts at the end of the development cycle or sprint, which often impacts sprint commitments and delivery dates. The best way to combat this is by incorporating testing into the development process and automating everything that can and should be automated, such as unit, regression, and performance testing.

There are great static analysis tools that can be integrated into the developers' pipeline to validate the quality, security, and unit test coverage of the code before it even gets to the testers. It's also important not to overlook tedious and unnecessarily manual processes, like test data gathering and test results analysis.

Even when all the technical bottlenecks are addressed, miscommunication and lack of transparency can completely break the flow and become huge blockers in an otherwise continuous and streamlined process. Defining ownership, understanding your audience, creating clear and simple messaging, and managing expectations are critical tools in breaking the communication barrier and moving toward a collaborative, agile organization.

3

It's a Pivotal Time for DevOps

5

6 Steps to a Successful DevOps Adoption

8

Engineering Architecture Systems for a Faster Build

9

The Essential Role of QA in Digital Transformation

11

Build the Right Things and Build Them Fast: Accelerate the Continuous Delivery Pipeline

12

Finding the Bottlenecks in the Agile and DevOps Delivery Cycle

13

Insight from around the Industry

14

Additional DevOps Resources

# Insight from Around the Industry

3

It's a Pivotal Time for DevOps

5

6 Steps to a Successful DevOps Adoption

8

Engineering Architecture Systems for a Faster Build

9

The Essential Role of QA in Digital Transformation

11

Build the Right Things and Build Them Fast: Accelerate the Continuous Delivery Pipeline

12

Finding the Bottlenecks in the Agile and DevOps Delivery Cycle

13

Insight from around the Industry

14

Additional DevOps Resources

"As far as the 'speed versus quality' trade-off, with modern software practices, you should not have to make a choice. By automating and building quality into your development and delivery process and 'shifting left' your testing, speed increases and quality improves. DevOps has proven that speed and quality are not mutually exclusive."

» *Mark Levy*

"The thing that we're finding now is, as we move towards continuous integration, continuous delivery, we actually become better managed and less risky by decoupling our software, being able to move things into production early and often. What we're doing now is we're able to automate a lot of our controls and make them part of the process."

» *Adam Auerbach*

"Agile and DevOps are about utilizing various tools to continuously improve the process, sometimes failing but eventually succeeding achieving small wins and collecting feedback along the way. As long as we are passionate about solving the problem, and have an awareness of the available tools, we will be able to move towards that coveted future state of continuous delivery."

» *Tanya Kravtsov*

"The big thing that I think has changed over the past year or so is that people are no longer saying, 'If we ever get to the point where we're doing DevOps.' They're at least now admitting that it looks like DevOps is the right way. It's the same thing we saw with agile a couple years back."

» *Gene Gotimer*

"Infrastructure support's gotten a lot easier. It's a lot easier doing things like DevOps now and being able to build realistic environments automatically and programmatically so that you can realistically test things in parallel different environments to make sure things haven't broken in a continuous deployment kind of scenario."

» *Neal Ford*

"I think the first 'Oh, crap!' moment that everybody has when they start thinking about DevOps, continuous delivery, continuous deployment, and all of those kinds of things is, 'What do you mean you're going to check in code fifteen minutes into production? What if it doesn't work? What about security? What about manageability? What about accountability?' I think those are very right things to worry about and, obviously, one of [Ops'] main goals in life is to keep the lights on and money coming in."

» *Anders Walgren*

*As long as we are passionate about solving the problem, and have an awareness of the available tools, we will be able to move towards that coveted future state of continuous delivery.*

"With continuous delivery, you can nudge in different directions or run A/B tests, see how the users like [the product], and move in the direction. Now we're making product decisions based on data, not based on speculation or conjecture."

» *Jeff "Cheezy" Morgan*

"Automation which used to happen at the end of the testing lifecycle is now a thing of the past. Now we are talking about how progressive automation, or holistic automation across the lifecycle, can enable the development teams to accelerate the process to integrate and release teams to accelerate the process to deploy the code on a continuous delivery model."

» *Shankar Konda*

# Additional Resources

## MORE INFORMATION FOR SOFTWARE PROFESSIONALS



DevOps is becoming essential for faster software development and deployment, so keeping up with expert advice and discussions on ways to navigate and improve your adoption and methods is key to staying ahead of the curve. DevOps To Go, a monthly TechWell newsletter, includes the most recent DevOps articles, interviews, and resources and is delivered for free to subscribers.

CLICK  
HERE

### NARROW YOUR SEARCH TO A SPECIFIC TYPE OF RESOURCE:

#### TechWell Insights

Find stories on DevOps, agile, testing, development, and more, written by software industry professionals. New stories are added each day, so **click here** to read the latest or sign up to receive the weekly newsletter.

#### Agile and DevOps Articles

From the early days of agile adoption to the latest DevOps technology, AgileConnection articles span the evolution of today's software development environments. **Click here** to access this invaluable collection of agile development and DevOps resources.

#### DevOps and Testing Articles

The DevOps push for continuous deployment requires testers to stay on high alert. **Click here** to read the latest articles about how DevOps affects testing and how testers are finding their place in a DevOps team.

#### Better Software Magazine Articles

*Better Software* magazine is a digital quarterly filled with expert analysis, how-to articles, and real-world case studies covering all aspects of software development. **Click here** to join StickyMinds and access DevOps articles from *Better Software* magazine.

#### Interviews

Each year, TechWell interviews dozens of software professionals including well-known thought leaders, seasoned practitioners, and respected conference speakers. **Click here** to read, listen to, and watch interviews with DevOps experts.

### Conferences providing innovative ideas & solutions for DevOps Professionals

DevOps **EAST**

A TECHWELL EVENT

DevOps **WEST**

A TECHWELL EVENT

**STICKYMINDS™**  
A TECHWELL COMMUNITY

See Past DevOps TechWell Conference Presentations



SQE Training offers training and certifications to get a more in-depth understanding of the topics you need to advance your career. See DevOps training opportunities here.