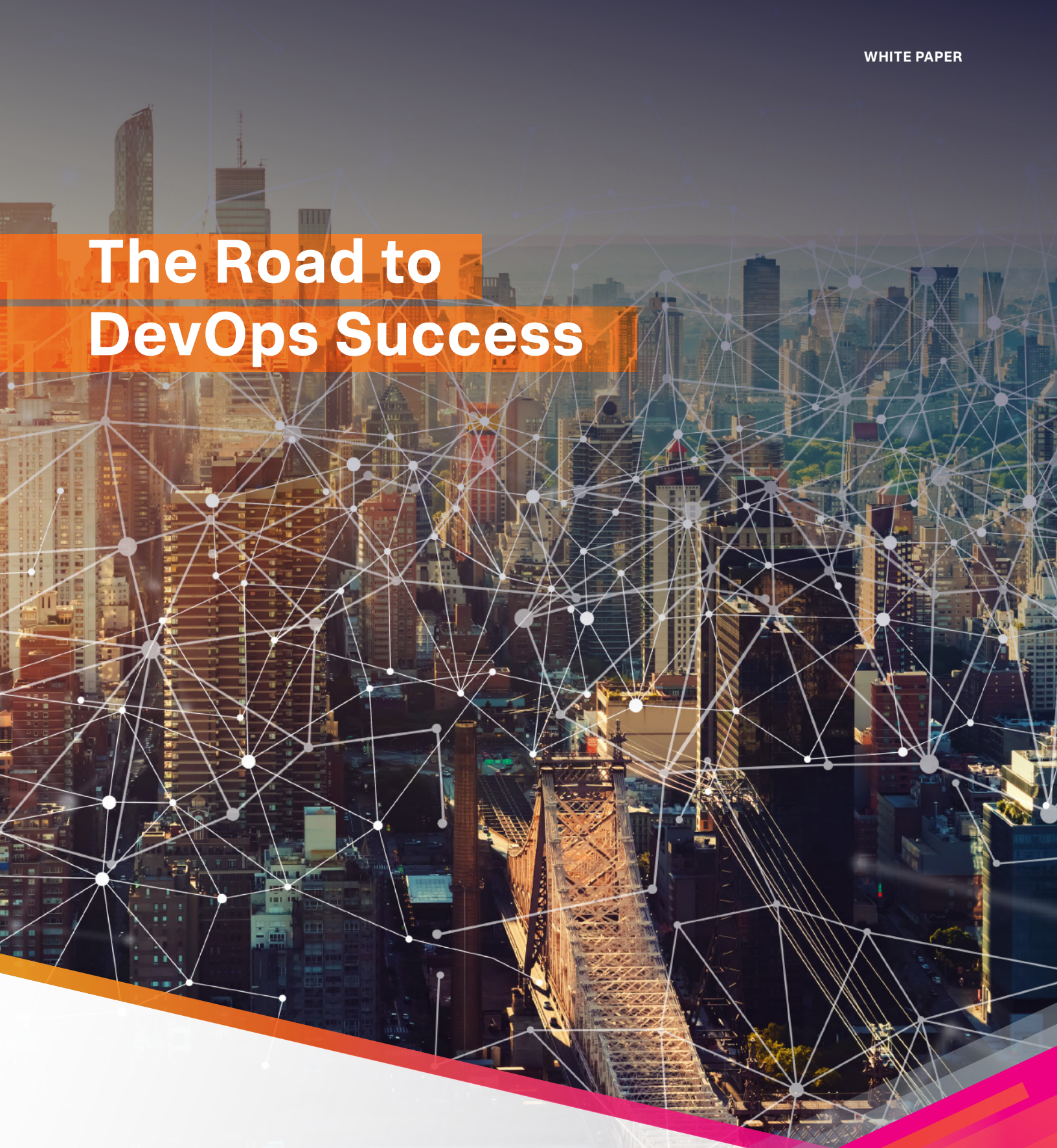
WHITE PAPER

The Road to DevOps Success



Achieve DevOps Success by moving from data collection to answers

splunk>
turn data into doing™

The digital economy moves at internet speed. Accordingly, organizations must deliver and refine new software and distributed services faster than ever — with agile, iterative software development and delivery. Frequent, smaller updates empower developers to rapidly try new features.

Frequent updates also help to:

- Keep apps fresh and top-of-mind with customers.
- Enable quick responses to threats and changes to operational needs.
- Limit the risk of breaking something critical.

Rapid app release cycles are common in digital businesses, where organizations are being reshaped by technologies like cloud services with service-oriented workloads, mobile apps and social communication channels. These cycles have prompted new architectures, design patterns, application stacks and continuous delivery release trains.

DevOps is a full lifecycle approach, based on a culture of collaboration. It is the primary way modern companies develop, deliver and support applications. Organizations that adopt DevOps move faster, enabling them to keep up with customer demands in an ever-changing digital world.

Yet, the hopes and dreams of DevOps devotees are often dashed on the rocks of organizational reality. DevOps doesn't always live up to its promises, often due to technological and bureaucratic complexity. Creating an agile, responsive, data-driven culture and supporting processes is easier said than done. While there are many challenges along the way, infrastructure and application complexity, application telemetry, and data analysis are frequent blockers to achieving DevOps excellence.

Data goes beyond logs. It comes from applications, cloud, infrastructures, containers, mobile apps, databases, the IoT, streaming network data and more. The activity and performance of all apps, systems, customers, APIs and industrial systems create logs, metrics, traces and events.

Achieving DevOps excellence requires more than just conventional performance monitoring. It requires analysis of all of this data to make sure that developers and operations folks can focus on business success and not firefighting. They can find answers in the context of the entire technology stack in real time to quickly understand how users are impacted, while avoiding business-impacting issues.

The reality of DevOps and the Eight-Phase Cycle

DevOps introduces agile development methods into service-oriented ephemeral workloads and operations focused on constant iteration. Code is defined, tested, deployed, observed and measured. When problems are encountered, the code is modified and the process starts again. Rinse and repeat.

"Code" includes the application code, of course, but also declarative architecture, network configuration, monitoring, etc. The entire process requires speed, agility and an organizational mindset that embraces incremental improvements and fast failure — not big bang software releases.

The key to a successful DevOps feedback loop is data — measuring and analyzing usage, performance, errors and key business metrics. The flexible nature of observability platforms to ingest, report and provide AI-assisted answers on any type of data makes them well-suited to DevOps processes.

Complexity is the enemy of agility, and agility is foundational to DevOps. A typical DevOps process includes eight steps with different tools used at each stage (Figure 1). Since these tools are often open-source or point products, there's little to no integration between them, creating overlaps and inefficiencies that can cloud visibility and hamper agility. This lack of agility means that issues exist for longer than necessary, slowing you down and costing you money.

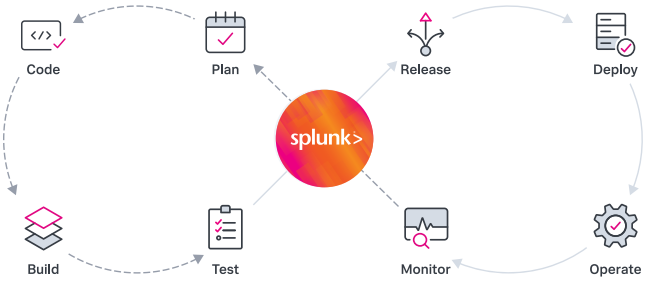


Figure 1: Major steps of a typical DevOps process

The infinity symbol is a typical depiction of the stages of DevOps, since it nicely illustrates the two intertwined classical owners of the process. The left side deals with the software development and testing cycle (the "dev"), and the right represents the deployment and operations

cycle (the "ops"). Yet each step in the cycle represents distinct product markets for tools, where a dozen or more products are commonly used in each stage.

This abundance of choice is great for developers and operations teams because each can choose products that suit its needs and preferences. But the resulting stew is a nightmare for project managers, DevOps leads and business execs. A mix of DevOps systems, each exhausting data in isolation, obscures visibility into the overall process, resulting in tool sprawl. Additionally, developers and operations teams do not use a common language or troubleshooting flow, extending the MTTR during outages.

This lack of common tooling often means there's no way to tell if "fixes" solve the root problem or merely provide a temporary patch. If teams lack complete visibility into the entire code development cycle, release and usage data, they can't validate app quality, performance and security. For example, security monitoring software can identify a vulnerability, but without a means of tracking the problem back to its source software module, the vulnerability cannot easily be fixed.

Additionally, when organizations lack visibility into DevOps metrics, they can't achieve the agility needed to try new ideas or iterate in response to changing business conditions. The result is poor quality, which can lead to customer dissatisfaction and churn.

In DevOps, disjointed build pipelines result in multiple, time-consuming handoffs throughout the application delivery process. These handoffs risk introducing redundant features and lowering efficiency through task-switching that drains everyone's resources. On average, it takes 10-15 minutes to resume editing code after a single interrupt.*

Requiring developers to context-switch in regular troubleshooting is a productivity killer. A disconnected delivery workflow also means impaired collaboration (with cross-departmental gaps by teams using different terminologies and having different viewpoints on the overall project). Together, the friction in communication and information sharing undermines trust and cooperation, which is counter to the core premise of DevOps.

Without a platform to provide data-driven feedback, DevOps is unlikely to be successful.

Realizing the full value of DevOps

DevOps is a buzzword for most executives, and a lack of familiarity often leads to unrealistic expectations. They have high hopes for what it can achieve, but lacking a proper understanding of the principles, process and challenges sets them up for disappointment.

Without an observability platform to provide data-driven feedback, DevOps is unlikely to succeed. The only way organizations can avoid this pitfall is to use metrics to measure, report and demonstrate the success of DevOps while using data-driven insights to optimize and improve it over time.

The true value of DevOps does not come from merely adopting a buzzword — it comes from a shared language and approach to software quality and delivery.

DevOps in practice

Successful companies are tuned into customer needs. Being tuned in requires an observability platform that provides constant feedback to help all areas of an organization: IT, app developers, line of business execs, security teams, auditors and others. By providing fact-based insights and decision support, an observability platform enables businesses to move swiftly to save costs, embrace good ideas and eliminate bad ones.

To do this correctly, you require data, tools and processes that provide all stakeholders with continuous insights into all of your workloads. DevOps lets you respond to these insights faster, across business units, applications and overall IT operations.

In practice, successfully coupling a DevOps model with comprehensive, AI-assisted analytics provides three key benefits: faster application delivery, better application quality and better business quality.

* http://blog.ninlabs.com/2013/01/programmer-interrupted/

Key capabilities to deliver results faster

Achieving agility and customer satisfaction requires DevOps teams to have four key capabilities:

- **Observability:** Every component of the DevOps build and delivery workflow must be instrumented with the data collected into a master repository for further analysis. Without data, it's impossible to understand and remediate issues.
- **Iteration:** Code fixes and improvements must be rapidly identified, triaged and developed using data correlated throughout the tool chain to provide deeper insights.
- **Collaboration:** Rapid delivery requires that DevOps teams are on the same page, use the same data and take action based on the same measurements.
- **Optimization:** Leaders must constantly strive to improve the process by making fact-based decisions. Process optimization requires data-driven answers to questions such as:
  - What is our rate of "idea-to-cash," our end-to-end throughput to profitability?
  - How long is each phase of our delivery pipeline?
  - How much time do teams spend writing, testing or reviewing code?
  - Which development teams are the most productive?

The benefits of data-driven DevOps flow directly to the bottom line — improving the business through greater efficiency and developer productivity, faster application delivery, lower costs, higher customer satisfaction and greater revenue.

Create better applications with real-time observability insights

Data from across the DevOps tool chain provides information that enables a proactive response to issues early in the development and testing cycle. With observability, developers and operations teams can have answers to problems before customers and users experience the frustration of broken features and crashing apps. This proactive response requires a common observability platform that everyone, including DevOps teams, can use as a single version of the truth — comparable to how source code control systems empower developers to consolidate and share work results.

Instrumenting and analyzing the entire DevOps process reveals the actual performance, usage, and error data critical to improving the end product and overall delivery process. A common data platform enables information to be correlated across tools and infrastructure and proactively alert to problems. For example, correlating data from code checks with performance monitoring systems can reveal issues before users file bug reports.

Consolidating data from throughout the DevOps delivery cycle requires a platform that can ingest data in real time from across tools used in the eight-phase cycle described earlier. Development tools are constantly changing, and so is the data.

Data is the raw material of DevOps measurement, but it must come from objective metrics that quantify whether code meets functional and operational specs and quality SLAs. Once code is deployed, teams need to understand what is happening within their applications as each release flows down the delivery pipeline. You can't understand your pipeline, or correlate pipeline events with application performance and end-user experience if you don't understand what is happening inside your application. Using real-time observability insights to analyze the entire application delivery cycle can help with:

- Consistent, measurable and trackable benchmarks for bug rates across development teams and code releases.
- Increased visibility of quality assurance metrics that identify problems before the production release.
- Making faster sense of ever-increasing distributed microservices, including the ability to identify where problems are when they happen.
- Optimizing cloud spend.
- Providing faster and more accurate alerting, directed troubleshooting and rapid insights with in-stream AI.
- Tighter integration of security within the DevOps delivery process, or what is increasingly known as DevSecOps.

Development teams can deliver more secure and compliant code by spotting and eliminating vulnerabilities early in a build cycle. The net result of real-time, data-driven insights from the application build pipeline enables security teams to identify dangerous behaviors, and alert the right team before the security issue is checked in to the codebase. Having an automated process to secure these behaviors makes security a first-class citizen in your development process, and is necessary for the modern age of applications.

Get better business results

The benefits of data-driven DevOps and observability flow directly to the bottom line — improving the business through greater efficiency and developer productivity, faster application delivery, lower costs, higher customer satisfaction and greater revenue.

Organizations that expand the use of a comprehensive observability platform beyond its traditional role in IT operations can expect:

- Real-time visibility of usage, performance, reliability, errors and security incidents for new application releases — not delays of hours or days as when using ad hoc reporting.
- Dramatically lower MTTR, up to 70% faster production problem resolution and 40% faster pre-production problem resolution.
- Greater efficiency through automated data ingestion and analysis. Developers and operations teams can focus on meeting the business needs and not wasting time on building and maintaining monitoring tools.

Ensuring DevOps teams stay aligned with business goals requires a continuous delivery process with frequent releases — measured and correlated with actual business results.

This fast release process improves the customer experience by delivering better performing and more useful code. Better code means happier customers, and happier customers are more loyal. Using data from an observability platform means customer satisfaction can be directly tied to code releases by analyzing metrics like application usage and sales. Other common observability business metrics include:

- The rate of customer sign-ups and downloads
- Revenue changes, including peaks and troughs associated with product releases.
- Changes in customer engagement and sell-through or cart abandonment.

Analytics quantifies and enhances DevOps benefits

Success in the digital world requires agility. This is achieved through an efficient DevOps release process, and observability platforms are needed to get the most out of a DevOps program.

DevOps practitioners must continuously improve responsiveness, collaboration, security and regulatory compliance. These metrics directly influence business reputation and customer satisfaction through the rapid development of innovative products. Yet, adopting DevOps can come with many challenges that are often the result of complex processes and tool sprawl. Having security data flow to one tool, systems performance data to another, and application data to a third can make correlation difficult. We must make it easy for business execs, IT leaders and development teams to understand the benefits of DevOps and how to realize them.

The secret to DevOps success is quantified validation. The adage, "If you can't measure it, you can't improve it" still applies, and a consolidated observability platform provides the ideal system for measurement and evaluation.

DevOps provides notable benefits, but only when executed correctly. It's a marathon, not a sprint. To fully "do" DevOps, you will need to change not only how software is developed, but also how it is released, operated, monitored and troubleshooted. It's a lot of change, but the benefits in productivity, developer and customer happiness, and release speed and reliability speak for themselves.

To become successful in DevOps today, you must embrace observability as an essential component of your software development practice. Observability helps you see and understand the state of each workload with context. With the complexity of today's cloud-native workloads, observability empowers DevOps teams to master the unknown unknowns with real-time answers for your applications — delivering critical insights to business stakeholders. To learn more, review the Splunk Beginner's Guide to Observability, or try Splunk Observability Cloud for yourself today.

Learn more about **Splunk for DevOps** or try Splunk software for **free**.

splunk>

Learn more: www.splunk.com/asksales

www.splunk.com

21-10758-SPLK-The Road to DevOps Success-107-WP