

Taking DevOps to the **Next Level**

Putting the Customer First



ca
technologies

TAKING DEVOPS TO THE NEXT LEVEL: PUTTING THE CUSTOMER FIRST

DevOps is no longer cutting-edge. Neither is continuous delivery, or even the shift-left paradigm.

Instead, these practices and principles have now been adopted by organizations, to an extent—and although the transitions remain incomplete in many cases, virtually all companies are aware of and are striving to implement DevOps workflows and organizational structures.

Yet not all companies approach DevOps in the same ways. To stand out from the crowd in the new, DevOps-dominated world and to achieve true value through the adoption of DevOps technologies and practices, organizations must leverage DevOps and related innovations in ways that allow them to create business value and optimize the customer experience.

Achieving this goal is challenging. Most DevOps tools and practices were conceived first and foremost as technical solutions to improve the workflows of the technical staff—software developers, IT ops staff, quality assurance engineers and so on. Transforming DevOps technical innovations into value for the business and customers requires taking DevOps to the next step.

How does an organization do that? The key considerations involve plugging the gaps that currently exist in typical DevOps delivery chains, especially in the realms of software monitoring and testing. Until organizations fully link and integrate the DevOps tools and workflows they have implemented, they will not be prepared to leverage DevOps innovations in ways that deliver true value for the business. The fruit of their DevOps labors will instead remain limited to achieving efficiencies from a technical perspective—which, although valuable, are not enough on their own to allow businesses to stand out from competitors in a world where virtually every company has embraced DevOps.

This white paper explains how to leverage DevOps tools and practices in a customer-centric way in order to build business value. It discusses the current challenges that most organizations face in extending DevOps innovations beyond the technical realm, identifies the gaps that companies must plug within continuous delivery chains in order to address those challenges, and explains how to make user engagement and the customer experience a prime consideration at all stages of the delivery pipeline.

The State of Continuous Delivery

Five years ago, the typical organization's software delivery chain remained primitive. It might have included a continuous integration (CI) server and perhaps some rudimentary, script-based release automation tools, but it otherwise relied heavily on manual workflows, hand-offs and interrupts.

The delivery chain was also likely functionally segmented, with each of its processes (such as development, testing and production management) performed in isolation by teams that did not communicate well.

This is no longer the case. The typical organization today has embraced DevOps and continuous delivery. It strives for constant collaboration and communication between its various teams, and it prioritizes the rapid release of software updates.

This is certainly an improvement over past practices. However, many continuous delivery chains still suffer from the following shortcomings.

Misalignment between technical and business goals

Most continuous delivery chains were designed and built to meet technical goals, with little consideration for overall business value or customer needs.

The typical continuous delivery chain enables rapid software delivery, automates tasks that would take engineers a long time to perform manually and supports the fast rollout of new features. It is driven by the collection of monitoring data that helps to optimize technical processes.



The State of Continuous Delivery

However, the overall customer experience receives considerably less attention in most delivery chains. Customer feedback about application performance and quality is typically collected manually and communicated on an ad hoc basis to DevOps engineers. Software updates are released according to schedules set by the tools that automate the delivery pipelines rather than by the needs of customers. The shift-left mentality encourages engineers to focus on pre-production monitoring and testing, which often wrongly comes at the expense of production application monitoring.

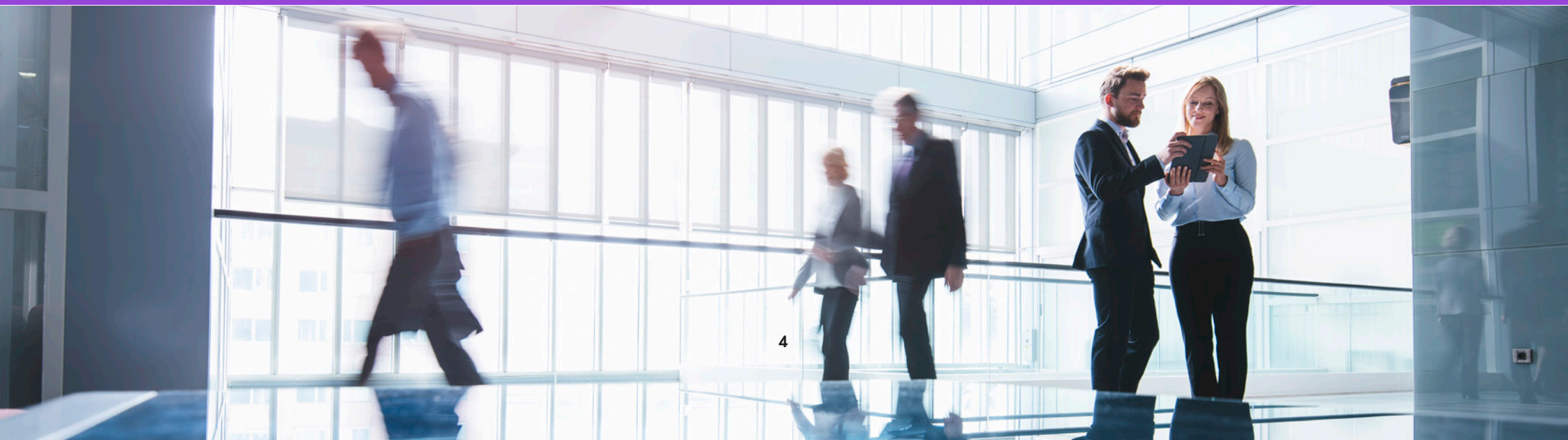
Narrow focus on cost

To the extent that delivery chains do address the need to drive business value, their focus is usually limited to cost. Continuous delivery pipelines are designed to use engineers' time most efficiently and eliminate bugs before they reach production and mitigate software bloat, all in the interest of allowing the business to operate more cost-effectively.

Cost-effectiveness is an important consideration, but it should not be the sole purpose of continuous delivery because it is not the only factor that matters in driving business value. The user experience, which is shaped in large part by application performance and quality, is equally important.

Lack of automated quality control and monitoring

For all that organizations have done over the past several years to automate most of their software delivery processes, certain key tasks (namely, software monitoring and quality control) are still performed on a manual or ad hoc basis, in many cases.



The State of Continuous Delivery

This is because organizations have typically tended to view these processes as the most difficult to automate. CI servers cannot fully help with quality control. Neither can release automation tools.

Despite the challenge of automating monitoring and quality control, no delivery chain can fully drive business value without automating these processes. Automation is essential for ensuring that the customer experience can be optimized as quickly and efficiently as software is delivered.

Non-continuous monitoring

The rapid rate at which software rolls down the continuous delivery pipeline means that organizations in many cases monitor software only in production. They fear that monitoring earlier in the delivery cycle would slow things down.

Sometimes, organizations respond to this limitation by adopting a "shift-left" software monitoring and testing strategy, which entails performing monitoring earlier in the delivery chain.

Both approaches carry risks because they limit monitoring to only one phase of the delivery chain. In other words, monitoring is non-continuous. Without continuous monitoring, which means collecting and responding to monitoring data and feedback at all stages of the delivery pipeline, organizations lack the visibility to ensure that the software they deliver into production meets users' expectations and that problems that emerge during production are addressed quickly.



The Business Risks of Imperfect Continuous Delivery

Although the shortcomings in continuous delivery chains described above may not cause technical issues that will delay software delivery, they will undercut the ability of the business to remain competitive.

That is a considerable problem in today's business environment. We live in an age when [40 percent of users will leave a website](#) if it fails to load within three seconds, and in which infrastructure performance problems can [significantly undercut](#) efforts to engage with customers.

In addition, consumers are [more worried than ever](#) about data privacy and software security. Because most end users lack the expertise to differentiate software performance and reliability problems from data security issues, any type of performance shortcoming can damage consumers' faith in a business's ability to manage data responsibly—and, by extension, can harm the brand.

If you think your business is not affected by these challenges because it is not in the IT industry, think again. As Satya Nadella [famously noted](#), every business is now a software business. Virtually every company, no matter what its industry, relies on software to engage with customers, drive marketing campaigns and more. Doing this effectively requires software that is not only produced quickly, but that is also tested and monitored thoroughly in order to ensure quality.

This is why a customer-centric software delivery chain is essential for protecting your business's reputation, as well as attracting and retaining customers. If the optimization of the customer experience is not part and parcel of your continuous delivery process, your business is at risk of losing what matters most: business.



Building a Customer-Centric Continuous Delivery Process

Creating a continuous delivery chain that prioritizes the customer experience is challenging. It requires thinking beyond the traditional goals and methods of DevOps, then adopting new tools and practices that support the integration of the customer experience into every stage of the delivery chain.

The following practices can help you achieve this goal.

Make quality everyone's responsibility

One of the original key innovations of DevOps was to make all members of the software delivery team responsible for helping to deploy and manage software. These tasks are no longer the responsibility of IT ops engineers alone.

The same approach must be applied to software quality. Testing engineers and on-call support staff who respond to application and infrastructure failures should not be the only members of the team who help to guarantee a quality user experience. Instead, everyone who participates in software design, delivery and management should have constant access to monitoring data in order to gain visibility into the quality and performance of the applications the team delivers. All team members should also be aware of the ways in which they can help to improve software quality through the various roles they play.

Rethink monitoring data

Traditionally, DevOps teams have leveraged monitoring data to identify bugs and failures within their applications. This is good, but it's not enough for optimizing the customer experience.



Building a Customer-Centric Continuous Delivery Process

Monitoring data should also be used to measure application performance trends and influence design decisions. For example, monitoring information from production environments can help to unmask poor coding practices that developers can change to improve application quality. For another example, by monitoring individual microservices and comparing monitoring data between them, organizations can identify which microservices are underperforming and need to be improved in order to enhance the overall application.

Maximize performance, not uptime

Maintaining 99.9 percent application uptime is a worthy goal, but it doesn't necessarily translate into an excellent customer experience. An application that is "up" may still be performing suboptimally, in ways that only a deep dive into monitoring data can reveal.

In addition, a fixation on simple uptime metrics can be highly cost-inefficient. Maintaining near-perfect levels of uptime is extremely expensive. If the investment required to increase uptime does not correlate with improvements in customer satisfaction, organizations waste significant amounts of money on an activity that does not increase business value.

For these reasons, organizations should focus on optimizing overall performance rather than the narrow metric of uptime. They can do this by collecting monitoring data that demonstrates how quickly an application is responding, identifies which services within an application are performing suboptimally, and helps map the relationships between services and infrastructure so that quality problems can be understood and addressed quickly, before they exert a significantly negative impact on customers.



Building a Customer-Centric Continuous Delivery Process

With this information, technical teams gain the insight they need to understand the relationship between application performance and the customer experience. They can then optimize their workflows to produce business value rather than spending their time pursuing technical goals that may not align with business priorities.

Collect data that reflects the customer experience

On a similar note, organizations should focus on collecting data that provides insight not just into the technical health of their application, but also into the way their customers experience the application.

Customer-centric data includes metrics such as the time required for a Web application to load, the number of users served per minute and variations in performance between Web, mobile and wearable applications.

Achieve continuous monitoring

As noted above, it is not enough to perform monitoring only before or after an application is in production. Monitoring needs to be continuous.

Organizations can achieve continuous monitoring by leveraging synthetic monitoring prior to application deployment in order to identify and fix problems before they reach customers. Monitoring should not stop there, however. Organizations should also monitor applications in production in order to address performance issues that arise once an application is in the hands of end users and gain insights that can help to troubleshoot problems revealed by testing in production.



Building a Customer-Centric Continuous Delivery Process

Indeed, monitoring in production and at all other stages of the delivery chain is the only way to gain the contextual information required to troubleshoot application problems effectively. For example, if a DevOps team delivers two versions of the same application, each with different functionality, production monitoring can help to ensure that both versions perform as intended. Testing alone cannot deliver this assurance.

Continuous monitoring is especially important when organizations adopt a continuous delivery strategy. In order to achieve the rapid, seamless deployment of software, DevOps teams must not only produce and test software quickly but also monitor it at all stages of the delivery chain; otherwise, undetected software quality problems may undercut the effectiveness of continuous deployment.

Understand all layers of the customer experience

In the DevOps world, your applications are no longer monolithic. Neither are your customers.

For this reason, your approach to monitoring and interpreting the customer experience should be able to contextualize monitoring data by associating it with the different customer personas, activities, use cases and expectations that your software must support.

For example, simply collecting data about overall application response time is not enough. You should be able to understand how response times vary across different geographies and whether slow responses are causing some customers to stop using your software—even if others find the response times perfectly acceptable.



Conclusion

Today, staying ahead of the competition and protecting the value of your business both require more than just technological efficiency. They demand a software delivery process that prioritizes the customer experience at every stage. Traditional DevOps tools and strategies were not designed to meet this challenge.

By adopting an innovative approach to software testing and monitoring, organizations can bring DevOps to the next level by making the delivery chain customer-centric. This approach is founded on continuous monitoring, the collection of context-aware monitoring data, and a focus on monitoring metrics that enable your team to assess customer satisfaction and improve the user experience on an ongoing basis.

Application performance monitoring and management from CA Technologies

CA Technologies' [application performance monitoring and management solution](#) can help. CA delivers rich, context-aware monitoring data that is presented cleanly and is easy to interpret. It supports monitoring at the various stages of the delivery chain to achieve continuous monitoring. Designed for use by the entire team, it removes silos and helps to make software quality everyone's responsibility.

In short, CA's application performance monitoring and management tools empower software delivery teams to extend monitoring far beyond mere metrics collection. With CA, you can optimize application performance, delight customers and drive business value.





< We invite you to [try the solution](#) for free today.



CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate—across mobile, private and public cloud, distributed and mainframe environments.

Copyright © 2018 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. This document is for your informational purposes only. To the extent permitted by applicable law, CA provides this document "As Is" without warranty of any kind, including, without limitation, any implied warranties of merchantability or fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document including, without limitation, lost profits, business interruption, goodwill or lost data, even if CA is expressly advised of such damages. CS200-355290_0418