Three Critical Capabilities Required to Achieve Continuous Delivery

ca technologies

Starting the Continuous Delivery Journey With DevOps

In the first chapter of this ebook series, we looked at trends impacting application delivery today—namely, the application economy and digital transformation. We also took a deep dive into the common challenges teams need to overcome to deliver new software innovations that support business goals and meet customers' expectations.

Business models are shifting constantly as companies find new ways to create and deliver customer value. Sustaining competitive advantage is increasingly dependent on IT organizations' ability to execute new digital transformation strategies that keep pace with the velocity of business change.

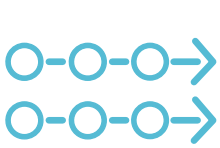
As a result, IT organizations are increasingly expected to drive differentiation, growth and profitability—and many are adopting DevOps practices to meet these mandates.

Overcoming chaos and complexity to enable Continuous Delivery is a fundamental goal of DevOps. In this chapter, we'll discuss how DevOps practices can help teams achieve Continuous Delivery by working collaboratively across functional silos to build more, higher-quality applications faster—and quickly deploy them into production with less effort and fewer errors.

BUILD MORE, HIGHER-QUALITY APPLICATIONS FASTER—AND QUICKLY DEPLOY THEM

Introducing the Three Capabilities Required for Continuous Delivery

Achieving Continuous Delivery to meet digital transformation goals requires new thinking, processes and tools in specific areas of the software delivery lifecycle (SDLC). On the following pages, we'll explore the critical capabilities that IT organizations need to overcome their most challenging Continuous Delivery obstacles, including:



1.

Parallel Work Streams

2.

Quality Acceleration

3.

End-to-end Automation

Step 1.

Parallel Work Streams

In an environment characterized by composite application architectures, asynchronous development and Agile methodologies, it is common for teams to experience constraints, bottlenecks and delays as they wait for dependent systems and services to become available.

However, digital transformation initiatives are increasing the pressure to crank out more builds faster—which is only possible when constraints are eliminated, so work can be executed in parallel.

Enabling teams to quickly simulate unavailable or constrained systems, environments and test data eliminates dependencies between project tasks and teams. As such, work can be done in parallel rather than sequentially. What's more, it reduces infrastructure costs by eliminating the need to duplicate production and test environments.

For example, teams can accelerate time-to-market by making it easy to publish, secure and control access to APIs used to assemble apps. Working prototype APIs can be published and connected to virtualized backend Web services, so teams can use them as soon as they're available, rather than having to wait until completion.

Removing critical dependencies from the development process ensures teams have what they need, when they need it. When constraints are eliminated and teams work in parallel, both **work and wait times are reduced dramatically**—enabling teams to keep pace with digital transformation goals.

Step 2:

Quality Acceleration

When development teams are cranking out builds as fast as they can, the QA team's workload increases as well—even as project timelines are being compressed. In addition, eighty percent of application testing today is manual—and often ineffective—because the available test data is too simplistic and does not represent the real world.

When workload and timeline issues pile up and threaten project milestones, it's common for test cycles to get squeezed—which leads to teams testing too little, too late.

To keep pace with Agile development cycles and business and user expectations, test teams must:

- **Reduce the manual effort required** for designing and executing unit, functional, regression, integration, load and performance tests
- **Speed up the testing process** by creating test plans automatically
- **Ensure rigorous testing** by provisioning the right data to the right place, at the right time
- **Find defects early** in the cycle when they are easier to fix, and create less re-work
- **Maximize testing coverage** and optimize testing efficiency to reduce costs without compromising quality
- **Ensure that stable test environments are available** across all stages, so that testing shifts left and begins earlier

When teams have access to the test data they need and the right automation capabilities, rigorous **testing can "shift left" to occur early and often**. This means defects and errors are discovered long before an app is deployed to production, so you can reduce re-work, keep projects moving at the speed the business demands—and ultimately ensure an excellent customer experience.

Step 3.

End-to-end Automation

Application delivery is an increasingly intricate process that requires collaboration across development, test, release management and operations groups—each with its own set of favorite tools.

While these groups may be efficient and automated within their individual silos, they often still rely on manually created scripts, mocks, stubs, etc. to be the glue that holds everything together. As their tools change and evolve, these manual assets must be recreated—a time-consuming effort that can negate the benefits of automation silos.

What's needed is a fundamental rethink of what it means to be automated. Rather than automating individual silos of the SDLC, teams must find a way to automate the entire release process as a unified, end-to-end, application delivery pipeline.

They can do this by:

- **Standardizing and automating release processes and workflows** that can be reused across multiple deployments
- **Automatically promoting releases** across the SDLC, from dev to test to production
- **Integrating best-of-breed solutions** and orchestrating them as part of an automated, continuous delivery pipeline that eliminates hand-offs and streamlines the entire process
- **Consolidating release data and activities** into a single UI for a unified view of the entire release process
- **Centralizing control** of the entire application lifecycle project, from requirements through production and deployment
- **Promoting collaboration** between development and operations teams by establishing continuous feedback loops of communication

---

End-to-end release automation across the SDLC reduces manual effort and errors, alleviates complexity and improves productivity, so you can streamline and **accelerate continuous delivery of business innovation**.

The Business Benefits of Continuous Delivery

Keeping pace with today's business and market demand for innovative, high-quality applications can present significant challenges, particularly if your application delivery systems and processes were designed to push out one or two releases a year.

The goal of Continuous Delivery is to overcome these challenges in order to get high-quality, innovative apps to market before your competitors. Not every organization needs to actually deliver applications continuously, but being able to deliver what the business needs, when it is needed, is a huge competitive differentiator.

For example, CA customers that have begun implementing Continuous Delivery solutions have experienced numerous benefits, such as:

UP TO **15x** Reduction in manual effort required for application release, **increasing productivity while lowering costs**[1]

UP TO **90%** More defects detected earlier when they're easier and less costly to fix, **improving quality and the customer experience**[1]

UP TO **50%** Reduction in the typical application development schedule, **accelerating time to market**[1]

[1] Based on actual Customers that have renewed with CA as of a 2014 review of CA Customer database.

What's Next?

In the final chapter of the series, we'll examine some of the tools that can help you accomplish the three steps discussed here, and show why CA is your trusted advisor to help you achieve your Continuous Delivery goals.

Discover your Continuous Delivery strengths and weaknesses with this **DevOps Assessment**, and receive actionable guidance you can use to start tackling your challenges 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. Learn more at **ca.com**.

CS200-132559-2

ca technologies