Continuous Testing

The Final Frontier of Continuous Delivery

ca technologies

In order to achieve true continuous delivery, organizations must rethink the way they carry out quality assurance (QA) today. This means rewriting the role that QA professionals play within their organization, striving for automation at every turn and rebuilding test harnesses to accommodate smaller, faster software builds.

CONTENTS

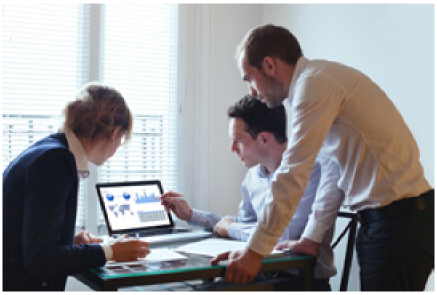
Effective Competition Depends on Continuous Delivery of Quality Software

In today's application economy every company is a software company, no matter what industry it is in:

- **Shipping companies** depend on logistics software to efficiently route packages, arrange drivers and automate warehouses.

- **Retail companies** rely on software to manage inventory, engage with customers online and to give in-store associates the tools they need to answer customer questions on the spot.

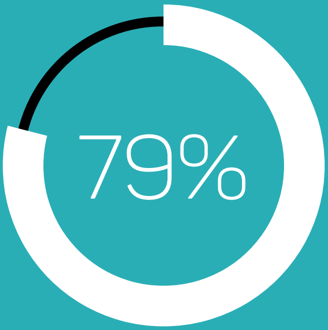
- **Marketing firms** lean on applications to gather consumer data and parse it, automate communication with prospects and effectively manage advertising campaigns.



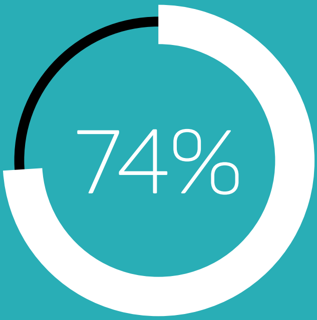
The examples are endless.

The point is that in order to compete today, every business must be able to quickly build and tweak software to adjust to always-evolving market demands. Ultimately, business success depends on faster development iterations while still maintaining the high quality of service expected by customers, stakeholders and end users.

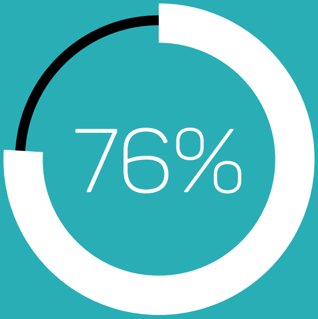
The digital transformations led by Continuous Delivery, DevOps and Agile best practices have very real business benefits:



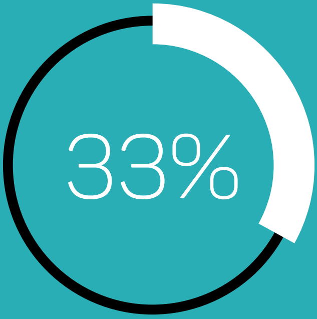
say increasing the frequency of software releases is a priority to meet business objectives



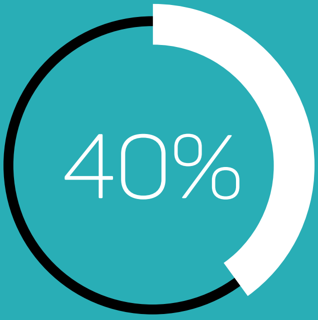
report improvement in customer experience



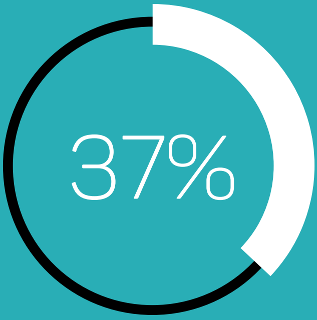
report improvement in digital reach



increase in speed to market



increase in customer satisfaction



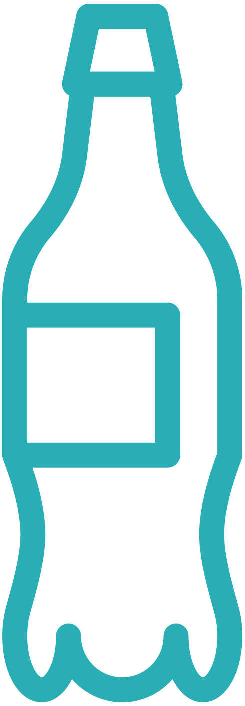
increase in new business revenue

Testing: Continuous Delivery's #1 Bottleneck

This goal of continuous software delivery has driven numerous paradigm shifts in IT thinking recently. The Agile and DevOps philosophies and patterns of operation have gone a long way toward helping many leading enterprises to speed up their software delivery processes.

However, even as organizations have moved away from waterfall development practices and worked to bring together developers and operational staff to streamline the delivery pipeline, they still struggle to truly achieve continuous delivery nirvana. One of the biggest impediments comes by way of an irritation best described in one word: testing.

QA and testing practices are turning out to be one of the number one bottleneck impeding the move to continuous delivery. Traditional testing procedures and technologies simply aren't equipped to handle the ever-increasing demand for faster testing. As a result, organizations are unable to truly achieve continuous delivery.



63% of organizations who have adopted DevOps processes report that their current QA processes are a bottleneck

28%
of organizations wish they could deploy hourly

14%
actually do so

You Can't Afford To Let Testing Go By The Wayside

So what's an organization to do?

While certain stakeholders may struggle with the QA process as it stands today, organizations can't afford to let quality concerns go by the wayside. As the expert consultants at Capgemini put it, "The consumer appetite to engage via digital channels is matched only by intolerance for any defects that might have an impact on their user experience and business outcomes."

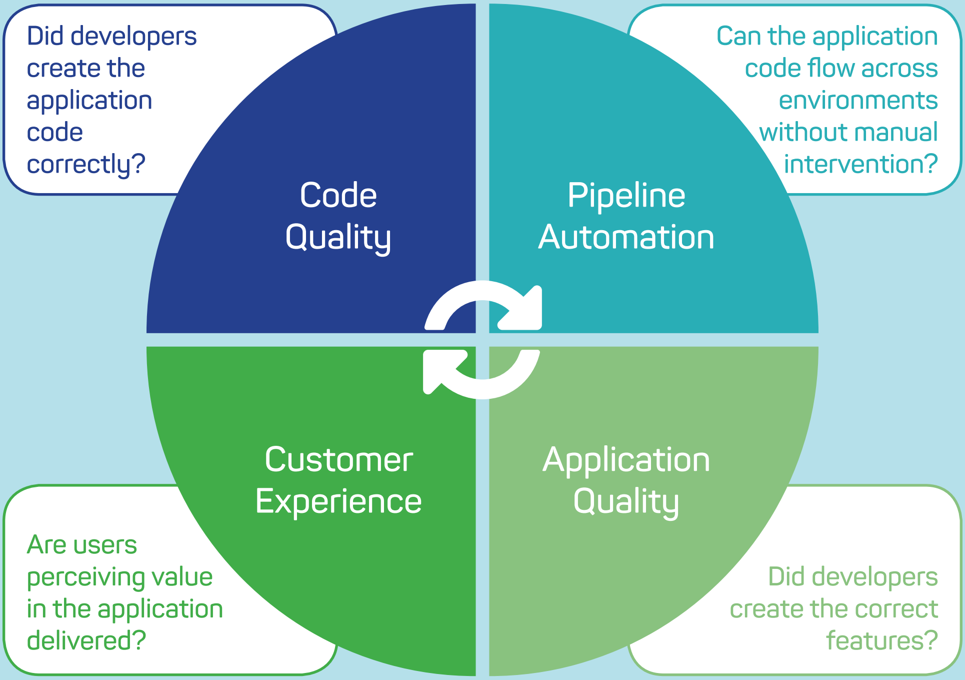
It's an opinion repeated again and again by IT experts.

"Today, it's not just about speed—it's about speed with quality. If testing is an afterthought, it happens between developing the software and going into deployment—and it feels like something that will stop you from delivering faster," writes Diego Lo Guidice of Forrester Research. "But a bug going into production in the age of digital is a real big problem. It will hit the news."

If organizations are going to speed up their delivery pipeline, they need to remember that the success of continuous delivery hinges on the effective employment of continuous testing.

The Promise Of Continuous Testing

To continuously deliver software, organizations need to make sure tests, procedures and processes around quality are designed to fit inline with the flow of development, integration and deployment. While continuous testing is dependent upon strategic automation, it is not synonymous with automated testing . A continuous testing process encompasses other QA activities that impact overall product quality. As such, continuous testing depends on four major pillars.


Did developers create the application code correctly?
Code Quality
Can the application code flow across environments without manual intervention?
Pipeline Automation
Customer Experience
Application Quality
Are users perceiving value in the application delivered?
Did developers create the correct features?

Currently, many organizations are unable to deliver these four pillars inline within their development cadence. The sad fact is that their QA remains shackled to old ideas and processes.

The Problem With Delayed Testing

The tradition of a long, manual testing process that's designed to be carried out when the application is 'done' is not very far in the rear-view mirror of most organizations. In fact, 22% of organizations still don't write tests until after development has completed its work.

But waiting to test applications until after the user interface is complete is a recipe for disaster. Often times results of functional, regression, user journey and performance tests for both applications and APIs will ultimately influence how the UI should be designed. Waiting to complete such tests until after the UI is done invites costly rework and poorly constructed user interactions. This threatens the very heart of an application's business goal: customer experience.

Fewer than
1 in 5 organizations
write new tests each time new code is checked in

Only
1 in 3 organizations
involve QA in planning requirements

Only
24% of organizations
break features into small testable requirements that are quickly iterated

Three Big Challenges of Continuous Testing

In the new continuous delivery era, there is no 'done,' and development work remains ongoing. Organizations that want to release more frequently need to rethink QA so that quality questions are asked from the requirements stage and tests are conducted every step along the way — particularly as smaller changes are made and code is checked in by developers.

Organizations need the capability to test early and often. Without that capability they will merely be practicing fast waterfall development rather than continuous delivery.

However, moving the needle on these metrics is easier said than done. Unfortunately, three big challenges tend to stand in the way of effectively achieving continuous testing.

Problem #1: QA's Role Remains Unchanged

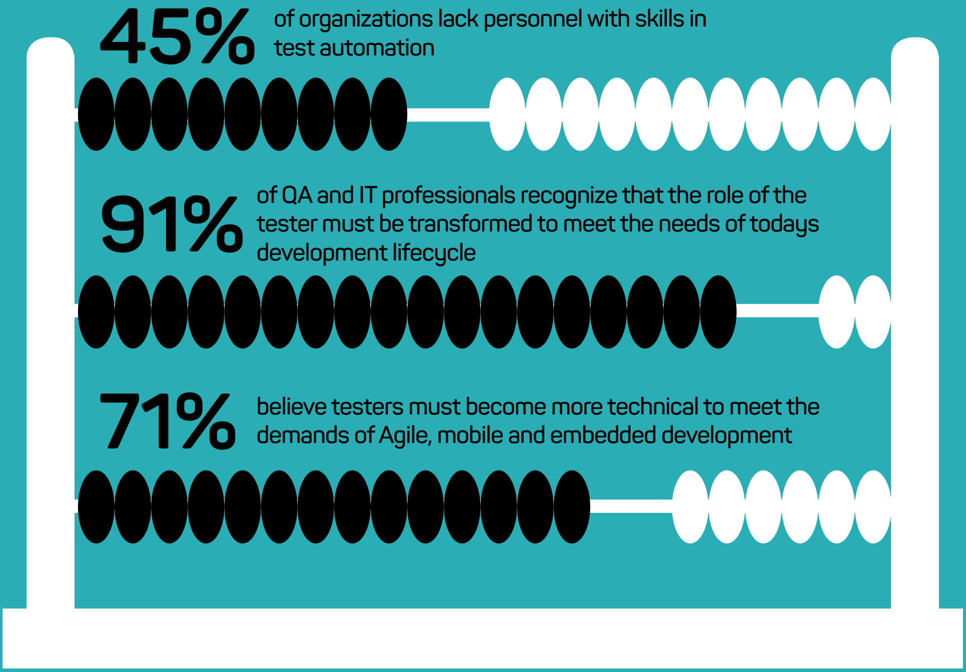
Problem #2: Many Tests Remain Highly Manual

Problem #3: Existing Tools Are Too Rigid

Problem #1: QA's Role Remains Unchanged

The QA role at most organizations remains unchanged compared to the reality of today's SDLC. For the most part, QA pros are primarily still thought of as testers. They're perceived as people who push the button on tests rather than advisors on how to automate and streamline testing processes.

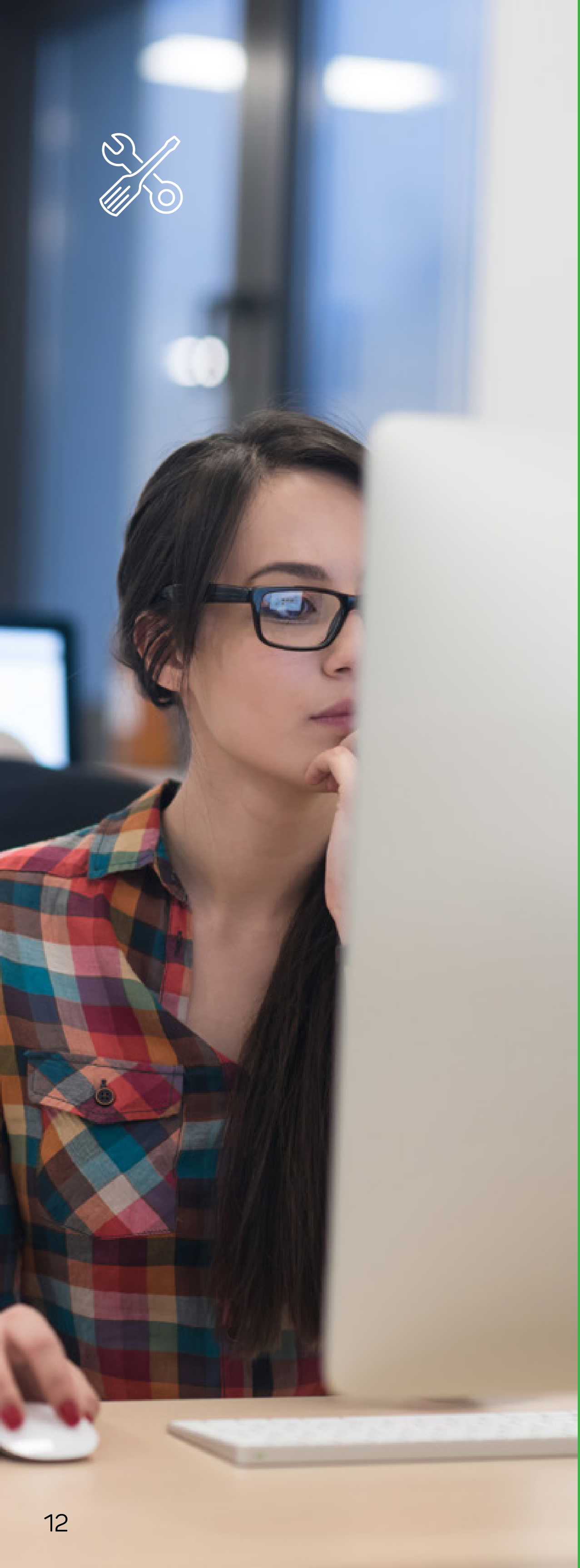
In many organizations, QA is also too far removed from development processes to move fast enough to enable continuous testing. Just as IT organizations needed to bring down the wall between devs and operations to enable continuous delivery, for continuous testing they need to remove the wall between the rest of the team and QA. Developers need to be deputized to be responsible for quality — but that doesn't mean QA goes away. Instead, organizations should be shifting the role from service providers to mentors, and from button-press testers to test-design engineers.



45% of organizations lack personnel with skills in test automation

91% of QA and IT professionals recognize that the role of the tester must be transformed to meet the needs of todays development lifecycle

71% believe testers must become more technical to meet the demands of Agile, mobile and embedded development



Problem #2: Most Tests Remain Highly Manual

Even if organizations are able to shift the testing process left — ensuring QA is inserted earlier in the SDLC, all is for naught if the tests placed inline are manual. With continuous testing the frequency of builds and code check ins will be such that no amount of testers can keep up with the volume of work if everything done by hand.

Organizations run into this time and again:

- The number one impediment to agile adoption is not enough automated testing
- Fewer than 1 in 5 organizations report that most of their testing processes are automated

Ideally tests should be able to be instantiated by developers, transparently and without hiccups. Instead, development flow is constantly put on hold while testers hit the pause button to conduct manual tests. This is a major impediment that will squash continuous testing from the get-go.

Problem #3: Existing Tools Are Too Rigid

One of the reasons many organizations cling to manual tests is because they've tried and failed with automation efforts in the past. The trail of these failures is littered with the corpses of rigid test frameworks and automation tools that aren't well suited for continuous delivery.

31% of organizations claim that test environments and test data sets are not flexible enough to quickly test micro-services.

45% of organizations claim that they don't have the right automation tools

40% of organizations face challenges with test data and environment availability

35% of organizations claim that delivery methodology doesn't support test automation

37% of organizations have difficulties integrating the different automation tools together

Evolving Toward Continuous Testing

Quality is everyone's responsibility, but some teams are more responsible for the big picture than others. Ideally, QA pros should be the champions of quality to keep the fire stoked under everyone else's proverbial rear ends.

This can be done by seeking to fulfill three important duties in the organization.

- **Test Strategist:** developers may be writing test scripts, but QA should be advising and mentoring in them in how to effectively design tests. QA pros bring unique perspectives about product quality issues that many developers by trade would not.

- **Toolsmith:** QAs will be the custodians of the test framework. They should be evaluating tools, piloting new processes based on new tools, ensuring that organizations don't over-engineer test tooling, and ensuring test environments are easily provisioned and test data is easily available.

- **Risk Manager:** QAs ultimately should be designing tests not as pass-fail exercises but as a way to measure risk thresholds. Their job is to help the organization think about how changes need to be deployed when the level is risk isn't acceptable and building rollback procedures when problems crop up.

If developers are deputized to do a lot more of the day-to-day quality checks, QA pros are the sheriffs helping to determine the testing policies and procedures that keep everything safe.

Ultimately, the goal of all of these fundamental changes to QA is to enact a continuous feedback loop that shifts testing both to the left and to the right of the development process. Organizations that do continuous testing well understand that what tests well in the lab could still fail in production. Testing components early is crucial, but it is equally important to monitor performance from user transactions within applications once they go live.

Step 1:
Start Addressing People and Culture Issues

The goal is to move from a center-of-excellence mentality to center-of-enablement mentality. To establish enablement, quality experts should be plugged into the development process. For example, organizations should embed QA experts into sprint teams.

Elevating the QA role will require organizations to cross-train QA professionals and developers to breed a new intersection of skills suited to a more technical role. This means injecting people into the QA team with skills around: automation and development skills, vendor management, environment and configuration management and data management, and communication soft skills.

Evolving Toward Continuous Testing

Step 2: Getting Test Harness Right

Organizations will find that their work to shift people and culture will see much greater success if they're armed with the right tools to do their newly enhanced jobs.

This means that sprint teams need:

- Test cases to be planned and automated from the requirements stage;
- Testable data at their fingertips;
- Consistent test environments quickly available;
- Simplified performance testing; and
- API testing in an open platform to ease test build process.
- Visibility and automation of testing hand-offs to understand when tests are passed or when applications are in jeopardy.

Finally, they'll need to take a comprehensive approach keeping in mind that testing must be baked into the entire software delivery pipeline. This requires a comprehensive test harness — an automated test framework that's right-sized for the organization's business and SDLC needs, and one which all the moving pieces are well integrated. When done right, the test harness will be a key enabler of achieving continuous testing and, as a result, true continuous delivery.

How Continuous Testing Looks In The Real World: GM Financial Case Study

These steps toward continuous testing can make a huge difference for organizations seeking to improve their competitive edge in the market. To understand the benefits, let's take a peek at how continuous delivery looks in the real world at GM Financial.

GM financial is the finance arm of General Motors. The company has been undergoing tremendous growth in the past six years and leans heavily on the nimble work of its development shop to keep up with loan transaction volume. In 2011, the firm only had one line of business platform. Today it is up to 13 such platforms. As the company grew, it started to hit the wall with traditional development processes.

> ***"We really got to a sort of a crossroads as a development and operations organization. We just knew we couldn't go further doing things the way we were doing, which was basically, you know, long manual deployments, lots of steps, lots of handoffs, and essentially, it was creating all kinds of contention problems, all kinds of resource problems, risks to production because we have so much creep between platforms."***
>
> ***—Matt McComas, assistant vice president of critical application infrastructure at GM Financial***

The firm started to alleviate the issues with release automation, but quickly became aware that testing automation and continuous testing needed to play an important part of its transformation.

In the past, it had gotten away with pulling test environments and data from production and putting it into test environments. But the team realized that wouldn't work at the pace and volume of development work needed to stay current with market needs. For example, in order to generate test funding scenarios to run through its software it was taking 15 to 20 people at least one to two weeks to create that data manually. Similarly, the firm was struggling to provide responses from third-party connections to replicate their activity within testing environments, sometimes stalling processes by as much as four weeks.

So, the team endeavored to build what GM Financial executives called a "soup to nuts cycle" for continuous testing. This included:

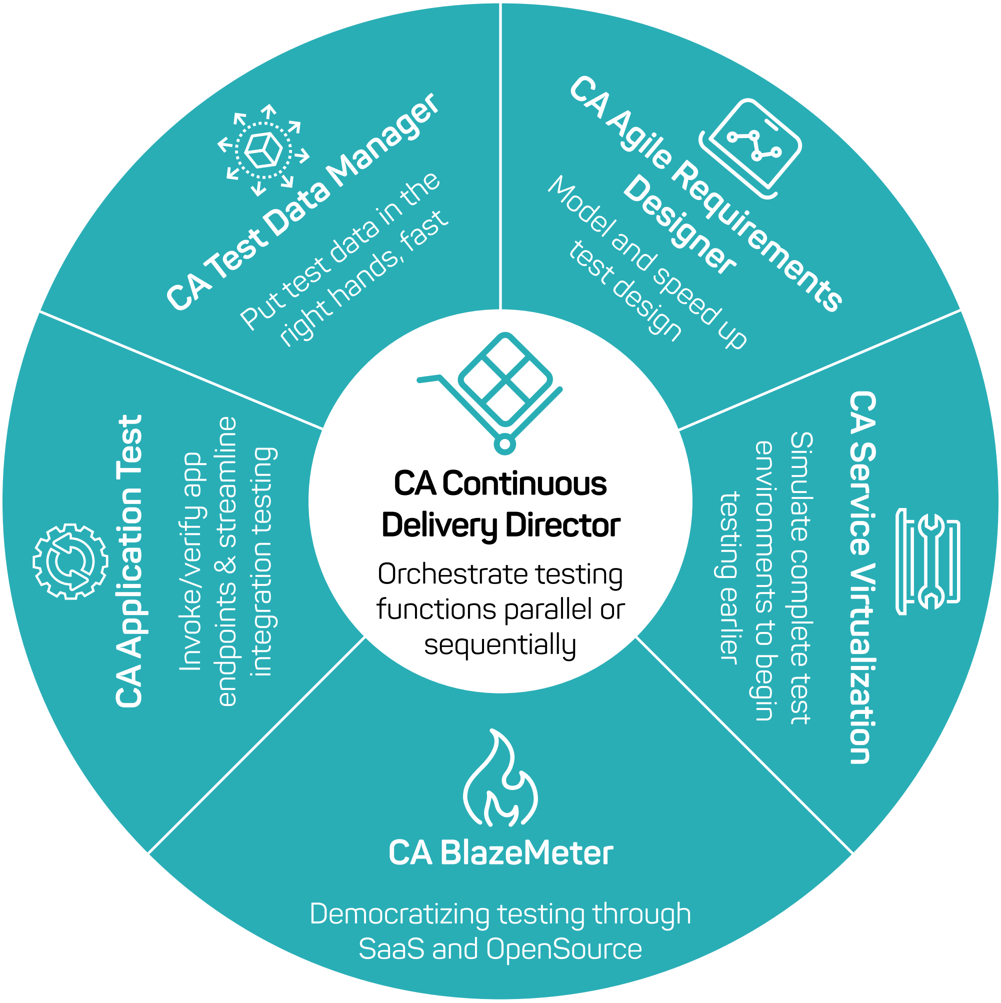
- **Test automation** for complete test progression through the test harness.
- **Service virtualization** to solve the third-party activity problem.
- **Test data management** to feed tests the data they need to run automated tests.

"We've got test data management and service virtualization, but mainly, test automation all coupled together, which gives us a pretty nice test harness to execute progression and other activities," McComas says.

CA Enables The Enablers Of Test Automation

If QA's new role is to be elevated from that of tester to enabler of quality, it will need effective partners to carry out its new mission. CA Technologies offers the optimum mix of testing tools purpose built to enable the enablers of test automation.

CA's Tools Help QA experts enable continuous testing in five key areas:



CA Continuous Delivery Director

Orchestrate testing functions parallel or sequentially

CA Test Data Manager

Put test data in the right hands, fast

CA Agile Requirements Designer

Model and speed up test design

CA Service Virtualization

Simulate complete test environments to begin testing earlier

CA BlazeMeter

Democratizing testing through SaaS and OpenSource

CA Application Test

Invoke/verify app endpoints & streamline integration testing

More importantly, when bundled together these capabilities offer a seamless and simplified experience for teams seeking to evolve toward continuous delivery.

One way to shift testing practices earlier in your software lifecycle is by using multi-layered visual models to specify requirements in a way where all ambiguity is inherently removed. With unambiguous and complete requirements, developers introduce less defects into their code and manual test cases, automated test scripts and required test data can be automatically generated based on the requirement, without manual intervention.

To help you start your continuous testing journey, we encourage you to get a 30-day free trial of CA Agile Requirements Designer to to begin implementing model driven testing, and to automatically create test cases based upon those user stories and business requirements.

Click here to get started with your 30-day free trial

http://www.ca.com
https://www.linkedin.com/company/ca-technologies
https://www.facebook.com/CATechnologies
http://twitter.com/CAInc
https://www.instagram.com/LifeAtCA/
http://www.slideshare.net/CAinc
https://www.youtube.com/user/catechnologies

ca® technologies

DevOps.com Custom Publishing