



Deliver Salesforce updates faster by tackling the #1 bottleneck in the software delivery process



“Every digital transformation is going to begin and end with the customer, and I can see that in the minds of every CEO I talk to.” That’s according to Marc Benioff, Chairman and Co-CEO of Salesforce. You might remember the quote from Benioff’s 2018 appearance at a Mulesoft conference, just days after Salesforce finalized its acquisition of the company.

What Benioff was getting at is this: Digital means different things to different businesses, but the one goal every transformation has in common is customer centricity. Becoming customer-centric requires enterprises to fundamentally redesign two things: how they provide value to their customers and how they operate. For many enterprises, Salesforce is at the heart of both of those changes.

Salesforce gives enterprises the ability to change fast – from streamlining operations to providing a foundation for entirely new digital customer experiences. With three releases a year, a highly customizable platform, and a set of tools that makes it easy to develop your own Salesforce applications, delivering value fast is what Salesforce is all about.

But the speed and ease with which users can make changes is a double-edged sword. While business users aren’t going to break Salesforce by making changes in production, they could easily break the customizations and new features your teams have built – which can both interrupt your ability to do business and introduce defects into customer-facing software that connects to Salesforce. It’s safer to build features in a development environment, test them in a test environment, and then push them into production. But the safe way of doing things slows down new feature delivery. If delivering value fast is what it’s all about, that’s a problem.

Cindy Breshears, Bluewolf’s Chief Transformation Officer, offered some helpful perspective on this challenge in a recent [blog post](#): “Create an organization-wide ‘factory’ that builds up capacity to keep Salesforce innovation operating, from scaling efforts to tackling new problems; otherwise, innovation will stagnate no matter how cutting-edge the strategy is.” It’s good advice, but it also sounds like a lot of effort.

Instead of siloing Salesforce delivery, leading enterprises are aligning Salesforce with modern delivery strategies across the enterprise application stack, and focusing on addressing the No. 1 challenge in the delivery process: speeding up new feature delivery without introducing risk.

This white paper outlines four strategies that can help you both achieve this alignment and tackle the unique challenges associated with delivering Salesforce innovation.

4 strategies for speeding up new feature delivery without introducing risk

1. Standardize on a scalable test automation tool

Research shows that it’s difficult to test thoroughly enough to reduce risk at the speed today’s business demands. According to [a survey conducted by GitLab](#), testing causes the most significant delay in the software delivery pipeline – even on DevOps teams. While test automation theoretically solves this problem, it

is notoriously difficult to scale. Recent [research from Tricentis](#) and the [World Quality Report](#) shows that a majority of organizations are still only automating about 20% of their total testing effort.

If you are one of the many organizations that's still doing 80% of testing manually, it's not all bad news. Look at it this way: addressing this bottleneck presents a significant opportunity for you to improve delivery speed. As the research indicates, past automation approaches have not helped application delivery teams keep up with increased demands of modern delivery. With growth in testing scope and scale, roadblocks emerge, from siloed knowledge and difficulty collaborating to overly complex toolsets that make it difficult to understand your progress.

To combat these challenges, a growing number of enterprises are opting for business user-friendly, no-code test automation solutions that can standardize testing processes across Salesforce and other mission-critical applications. A no-code approach can catalyze your transition from manual to automated testing by enabling both testers and business users to quickly automate testing.

A model-based approach further streamlines the process, with automation built from reusable building blocks, which can be repurposed across test cases and projects – something difficult to achieve with script-based tests, which must be updated every time the application changes. With model-based test automation, those same building blocks can be used as the basis for testing minor updates as well as major upgrades – like the one from Salesforce Classic to Lightning – so your releases are not delayed by a requirement to rebuild test scripts from scratch.

2. Consider a 'best of suite' or platform approach

Preferences are shifting away from the complex toolsets inherent to a best of breed approach, and towards a unified testing platform or suite that meets most testing needs. Nearly 80% of enterprises now prefer this approach, according to research conducted for [The Forrester Wave™: Continuous functional test automation suites, Q2 2020](#).

"This is not just a data point specific to functional test automation tools," said report author Diego LoGiudice, Forrester Vice President and Principal Analyst, in a recent [webinar](#). "It's an overall trend in the continuous delivery lifecycle," which today includes up to 80 different tools on average. "And that's why having a single, different best of breed tool for each one of them is something that very few want to deal with."

One of the largest health insurance providers in the U.S. chose to standardize test automation on a single platform after discovering they could not deliver updates quickly enough to keep up with their competition. Testing leaders created a testing center of excellence, adopted a model-based test automation solution, and outlined a plan to streamline testing processes across 18 enterprise technologies, from Oracle to Salesforce, and all the connections in between. The organization has set a target to increase test automation to greater than 80% across these applications in the next three years to support a robust CI/CD pipeline.

3. Simplify end-to-end testing

Delivering Salesforce updates quickly without introducing risk isn't just about Salesforce. The average Salesforce instance includes dozens of API integrations that connect it with end-to-end business processes that span from SAP to customer-facing software. Each new integration point becomes another potential point of failure – which can be particularly challenging to control when many of these applications are evolving in parallel. To more effectively reduce risk and protect the user experience, testing processes should be aligned across each of those applications.

While a traditional testing approach might focus on the functional areas of an individual application, this strategy alone is not sufficient. Testing the end-to-end business process is critical because any given part of an application might function differently when working in conjunction with another system than it does when working in isolation.

Given this situation, it's no surprise that Gartner lists end-to-end testing as a [critical capability for test automation software](#). In the most recent critical capabilities report, this use case was replaced with “cloud native/migration and SaaS software” – a recognition that a broad range of support is needed, whether you are migrating traditional workloads to the cloud or developing cloud native applications.

McKesson recently opted to simplify end-to-end testing, when a surge in demand made delivering high-quality updates on time difficult. End-to-end testing, previously carried out manually, crossed a set of 30 custom and enterprise applications, including Salesforce, Workday, and Snowflake, as well as a complex SAP environment. For many business processes, a single transaction flows from a customer interface through order entry, order delivery, and payment systems – with supporting processes in distribution, security, operations, finance, reporting and analytics, and more.

With a rapidly increasing testing scope, the team struggled to test just 30% of these end-to-end scenarios. By introducing model-based test automation, the team achieved 80% end-to-end test coverage, reduced testing timelines tenfold, and saw a 25% reduction in production defects.

4. Enable Salesforce admins and business analysts to break the testing bottleneck

Effective testing for any enterprise application requires that business users and technical resources are aligned. This is particularly true for Salesforce.

As Idea Science's Peter Drum pointed out in a [presentation](#) about a major Salesforce testing transformation at Inchcape, “You have to be able to focus on the business process to provide real value to the business. If all you're doing is breaking things down into very functional elements, then you're kind of missing the point.”

Inchcape, a global automotive distributor and retailer, was struggling with regression testing delays that made their monthly Salesforce release cadence nearly unattainable. Testing the regression suite manually – which included 500 test cases and growing – took the equivalent of 10 person days. The team knew automation was the answer, but they had not had much luck scaling with open source tools. Instead, they adopted a codeless solution that enabled their Salesforce users – who were intimately familiar with the most important Salesforce

processes but not with test scripting – to carry out test automation. With codeless test automation in place, the team is now able to complete regression testing in one day.

Salesforce makes it easy for business users to create their own customizations – no coding skills required. Because all of those customizations have to be tested, it's easy to end up with an enormous test suite. On the other hand, if business users and technical resources are not aligned, the team carrying out testing may not even be unaware of potential points of failure until they break in production. The answer to no-code customization is no-code testing. This makes it easier to align changes back to the existing test suite, or if technical resources are carrying out testing, to alert testers to the changes. In the latter scenario, a record-and-playback tool can enable business users to easily document business process changes that need to be tested.

Using an impact analysis tool to understand the impact of the changes made in each release can further streamline the testing process. Impact analysis can pinpoint where changes have been made, what processes they're likely to affect, and which tests will cover those changes and processes. This significantly cuts down on the amount of testing required, by pinpointing the most important automated tests to run.

Whichever solution you choose, one thing is clear: testing must be able to keep up with changes being made by both the business and your technical teams, as well as the updates that Salesforce releases.

About Tricentis

Tricentis is the global leader in enterprise continuous testing, widely credited for reinventing software testing and delivery for DevOps and agile environments. The Tricentis AI-based, continuous testing platform provides automated testing and real-time business risk insight across your DevOps pipeline. This enables enterprises to accelerate their digital transformation by dramatically increasing software release speed, reducing costs, and improving software quality. Tricentis has been widely recognized as the leader by all major industry analysts, including being named the leader in Gartner's Magic Quadrant five years in a row. Tricentis has more than 1,800 customers, including the largest brands in the world, such as Accenture, Coca-Cola, Nationwide Insurance, Allianz, Telstra, Dolby, RBS, and Zappos.

Learn more about Tricentis' Salesforce testing capabilities [on our website](#).