

The logo for ACCELOQ, featuring the word "ACCELOQ" in white uppercase letters on a black rectangular background. The letter "Q" is stylized with a green dot.

Getting Started with

Codeless AI in Salesforce Testing

A Beginner's Playbook

The Salesforce logo, which consists of the word "salesforce" in white lowercase letters inside a blue cloud-like shape.

PARTNER

Contents

Introduction to Salesforce Testing	3
Impact of AI and No Code in Salesforce Testing	6
Key Challenges in Salesforce Test Automation	8
Setting Up the Salesforce Test Automation Environment	11
ACCELQ - AI-Powered Codeless Salesforce Test Automation Platform	14



Introduction to Salesforce Testing

Salesforce, a cloud-based Customer Relationship Management (CRM) platform, has evolved into a cornerstone for businesses seeking efficient management of customer interactions and data. As organizations increasingly rely on Salesforce to streamline their operations, the need for robust testing processes becomes paramount. Salesforce testing encompasses a series of methodologies and practices aimed at ensuring the reliability, functionality, and security of applications built on the Salesforce platform.



Overview of Salesforce Testing and its Significance

Salesforce testing involves a systematic examination of Salesforce applications to identify and rectify potential issues. This encompasses testing various aspects such as data integrity, user interface functionality, and system integration. The primary goal is to ensure that Salesforce applications meet business requirements, operate seamlessly, and provide a positive user experience.

The significance of Salesforce testing extends beyond mere bug detection. It plays a crucial role in:

Ensuring Data Integrity Salesforce often handles sensitive customer data. Testing ensures the accuracy and security of this data, preventing potential data loss or corruption.

Mitigating Risks in Customizations Many organizations customize Salesforce to align with their specific needs. Testing safeguards against risks associated with these customizations, preventing unintended consequences.

Compliance and Security For industries with regulatory compliance requirements, Salesforce testing is crucial for ensuring adherence to standards and maintaining data security.

Optimizing User Experience thorough testing helps identify and rectify issues related to the user interface, ensuring a smooth and intuitive experience for end-users.

Supporting Continuous Improvement As organizations evolve, so do their Salesforce implementations. Testing facilitates ongoing enhancements and updates, allowing businesses to adapt to changing requirements.

Benefits of Automated Testing in Salesforce Development

Automated testing has become an integral part of the Salesforce development lifecycle, offering numerous advantages that contribute to the overall efficiency and reliability of applications. Some key benefits include:

Accelerated Testing Processes Automated testing significantly reduces the time required for repetitive and time-consuming testing tasks, enabling faster release cycles.

Early Detection of Defects Automated tests can be executed continuously, enabling the early identification and resolution of defects in the development process.

Resource Optimization By automating routine testing, manual testers can focus on more complex and creative aspects of testing, utilizing their skills more effectively.

Enhanced Test Coverage Automated tests can cover a broad spectrum of scenarios, ensuring comprehensive testing that might be impractical with manual methods.

Consistency and Reusability Automated testing provides consistent results and can be reused across different stages of the development lifecycle, promoting efficiency and reliability.

Salesforce testing, when coupled with automated testing practices, becomes a cornerstone in delivering high-quality applications that align with business objectives and meet the expectations of end-users.

Impact of AI and No Code in Salesforce Testing

As technology advances, Artificial Intelligence (AI) and No Code/Low Code approaches are playing pivotal roles in reshaping the landscape of Salesforce testing.



THE GROWING ROLE OF AI IN TESTING

Intelligent Test Case Generation

- ML algorithms analyze historical test data, application usage patterns, and system changes to generate test scenarios.
- This saves time and ensures better coverage of test scenarios.

Dynamic Test Data Management

- AI can generate realistic test data, ensuring that test scenarios are close to real-world usage.
- This is particularly beneficial in Salesforce testing where data integrity is paramount.

Predictive Analysis for Defect Prevention

- AI can analyze huge amounts of data to predict potential defects based on historical patterns.
- It identifies areas of high risk and assists in proactively addressing issues before they impact Salesforce.

Adaptive Test Automation

- AI-driven test automation tools adapt to changes in the Salesforce application, making them more resilient to modifications and updates.
- This helps maintain effectiveness of automated tests in dynamic development environments.

THE GROWING ROLE OF AI IN TESTING

Intelligent Test Scripting

- AI enhances the process of test script creation, analyze application elements/ interactions to automatically generate test scripts in plain English.
- This is advantageous in Salesforce testing, where applications are often complex and highly customizable.

Cognitive Test Execution

- AI-powered testing tools can mimic human-like cognitive abilities, allowing for more intelligent test execution
- Can understand complex scenarios, adapt to changes, and make decisions during test execution

Self-Healing Test Automation

- Enables self-healing automation, where the testing tool automatically adjusts test scripts to accommodate changes in the Salesforce application

Natural Language Processing (NLP) for Test Documentation

- NLP, a subset of AI, is employed for converting human-readable test documentation into executable test scripts.
- This streamlines the test creation process, making it more accessible to non-technical stakeholders involved in Salesforce testing.
- AI in Salesforce testing enhances the efficiency and effectiveness of test automation
- It introduces capabilities that were previously challenging to achieve.
- The synergy between AI and No Code/Low Code approaches contributes to making Salesforce testing more accessible to a broader audience, including those without extensive coding



Key Challenges in Salesforce Test Automation and Their Solutions

Salesforce test automation, while offering numerous benefits, comes with its set of challenges that organizations must address to ensure the effectiveness of their testing processes. Here are key challenges and practical solutions for each:

The Data Challenge

Challenge

- Data migration complexities, including ensuring data quality and mapping. Navigating Salesforce API data rate limits.
- Proper integration and testing of REST APIs is crucial for consistent and accurate data.
- Simulating real-time data for comprehensive testing.
- Integration testing challenges with adherence to protocols and standards.

Solution

- Implement a robust data migration strategy with a focus on data quality and mapping.
- Monitor and manage Salesforce API data rate limits during testing.
- Use test automation to thoroughly test REST APIs, ensuring proper import and export.
- Simulate real-time data scenarios in testing environments.
- Employ automation tools that allow flexible handling of HTTP methods, headers, parameters, and response codes.

Navigating Conflicting Codebases

Challenge

- Infinite customization and integration capabilities leading to conflicting codebases
- Multiple developers working on different modules result in diverse technology stacks.
- Dynamic object IDs increase the chances of failed tests.
- Anomalies across technology stacks need intelligent identification.
- Complexity in managing conflicting codebases during automation testing.

Solution

- Implement version control and collaboration practices to manage conflicting codebases.
- Develop intelligent automation scripts to identify and highlight anomalies across diverse technology stacks.
- Utilize test automation tools that can adapt to dynamic object IDs.
- Employ tools with advanced reporting features to highlight conflicts and facilitate resolution.
- Implement regular code reviews and communication channels among developers to minimize conflicting codebases.

Dynamic Tables

Challenge

- Salesforce's table-heavy design with dynamic rows and adjustable tabs.
- Challenges in selecting active tabs and performing actions using automation tools.
- Complications in ensuring proper functionality of commands like `driver.switch_to.window` and `send_key`.
- Integration testing complexities arising from dynamic tables and their interactions.

Solution

- Develop automation scripts that can intelligently interact with dynamic tables and tabs.
- Ensure that test automation tools can handle dynamic elements, such as checkboxes and commands like `driver.switch_to.window`.
- Implement thorough testing of dynamic tables and their interactions with test automation tools.
- Regularly update automation scripts to adapt to changes in dynamic table structures.
- Employ automation tools with robust handling capabilities for dynamic UI elements.

Executing Against Dynamic Content

Challenge

- Dynamic content in Salesforce applications without fixed IDs, names, classes, or CSS properties.
- Challenges in managing dynamic content, avoiding duplicate records, and ensuring accurate data during integrations.
- Salesforce's regular updates causing dynamic element locations to change
- Conventional approaches to detect fixed components of the locator not being effective.

Solution

- Develop custom coding approaches for locating and interacting with dynamic content.
- Implement strategies to handle dynamic content without relying on fixed IDs or traditional locators.
- Regularly update automation scripts to adapt to changes introduced by Salesforce updates.
- Utilize automation tools with robust support for dynamic content handling.
- Implement synchronization strategies to manage changes in dynamic content during integration testing.

Navigating Through Frames

Challenge

- Salesforce's frontend dynamic frames complicate access during testing.
- Most test automation tools face challenges in loading content independently within frames.-
- Effort-intensive process of finding the right attributes and scripts for nested or hidden frames.

Solution

- Utilize automation tools that support the independent loading of content within frames.
- Implement explicit instructions in automation scripts for actions within frames.
- Develop a systematic approach for identifying and handling nested or hidden frames.
- Invest in tools that simplify the process of finding attributes and scripts within frames.
- Regularly update automation scripts to adapt to changes in frame attributes and structures.

Shadow DOM

Challenge

- Salesforce's use of Shadow DOMs to encapsulate web component structures.
- Test automation tools struggling to recognize Shadow DOM components using traditional selectors (ID, Xpath).
- Testing complexities arising from the heavy DOM structure of Shadow DOMs.

Solution

- Implement custom coding approaches to recognize and interact with Shadow DOM components.
- Regularly update automated tests to account for changes in Shadow DOM structures.
- Invest in test automation platforms that support testing within Shadow DOMs.
- Collaborate with developers to ensure proper exposure and identification of components within Shadow DOMs.
- Consider leveraging specialized tools designed for testing applications using Shadow DOMs.

Setting Up the Salesforce Test Automation Environment

Setting up the Salesforce test automation environment is a crucial step toward ensuring efficient and effective testing processes. This involves configuring the testing environment and addressing prerequisites for automation tools. Here's a detailed breakdown:

Environment Configuration

- **Define Test Environments** Create environments that accurately reflect production scenarios to ensure testing aligns with real-world usage.
- **Configuration Management** Configure environments to support test automation processes seamlessly.
- **Isolation** Isolate test environments to prevent interference with production data and configurations.

Prerequisites for Automation

- **Compatibility Check** ensure that chosen automation tools are compatible with the Salesforce API versions in use
- **Credentials and Permissions** verify access to necessary credentials and permissions for the Salesforce environment.
- **Integration Compatibility** confirm that the selected automation tools integrate seamlessly with the Salesforce platform.

Choosing the Right Automation Tools

Selecting the appropriate automation tools is a critical decision. Consider the following factors:

- **Ease of Use**
Opt for tools that offer user-friendly interfaces and ease of navigation.
- **Scalability**
Ensure that the chosen tools can scale to accommodate the evolving needs of the Salesforce application.
- **Salesforce-Specific Features**
Look for tools that specifically support Salesforce features and functionalities
- **AI Capabilities**
Explore tools with AI capabilities to enhance testing efficiency and effectiveness.



Best Practices for Efficient Automation

By following these practices, your Salesforce testing framework will efficiently handle integration challenges, support omnichannel testing, and ensure continuous improvement.

- ✓ **Integration Testing Framework**
Identify integration patterns and address data, process, and virtual integration demands. Develop standardized test scenarios for each pattern.
- ✓ **Technology Stack Inventory**
Document the technology stack and interactions with AppExchange and custom apps. Create a matrix mapping interactions between Salesforce and each application.
- ✓ **End-to-End Testing**
Develop comprehensive test cases and automation scripts for end-to-end scenarios. Integrate automated tests into continuous integration pipelines.
- ✓ **Code Portability**
Place test code in the same class as the original code for portability. Validate tests during migration from sandbox to production.
- ✓ **Shift Testing Left**
Involve testers early in the development lifecycle. Develop automated unit tests integrated into the development workflow.
- ✓ **Comprehensive Documentation**
Document all test case scenarios and exceptions centrally. Update documentation regularly to reflect changes.
- ✓ **Continuous REST API Testing**
Involve testers early in the development lifecycle. Develop automated unit tests integrated into the development workflow.
- ✓ **Comprehensive Documentation**
Use a comprehensive automation platform for REST API testing. Generate diverse inputs, provide detailed error reports, and ensure code-coverage visibility.
- ✓ **Omnichannel Testing**
Develop channel-specific test cases for consistent user experience. Simulate realistic scenarios for omnichannel interactions.
- ✓ **Monitoring and Reporting**
Implement monitoring tools for real-time data flows and system interactions. Generate detailed reports and use dashboards for quick visibility.
- ✓ **Collaboration and Concurrent Testing**
Provide a shared testing environment for collaboration. Establish a feedback loop for continuous improvement.

ACCELQ

An AI-Powered Codeless Salesforce Test Automation Platform

ACCELQ is a leading Continuous Testing platform, deeply aligned with Salesforce to accelerate and improve the quality of your Salesforce releases significantly. The platform is a Forrester Wave Leader 2022 in Continuous Automation Testing Platforms.

Salesforce Release Alignment

Being an ISV partner, ACCELQ is aligned to Salesforce releases to ensure smooth Salesforce upgrades with robust Automation testing

Codeless Salesforce automation of Web, Mobile & API

Easily automates end-to-end flows in an integrated Salesforce environment from Web-API-Mobile. AI-driven no-code test automation on cloud optimized for Salesforce-specific Dynamic technology.

Accelerates Salesforce test automation

With pre-defined Codeless Salesforce test assets with real-time Salesforce release alignment.

Used by Internal Salesforce Teams

ACCELQ is used by internal Salesforce teams like the Pre-release Business scenario testing team.

On Salesforce AppExchange

ACCELQ is the only cloud-based Continuous testing and automation platform on Salesforce AppExchange.

ACCELQ

7.5x

Faster Test Cycle

72%

Lower effort On
Test Automation

3x

Acceleration in
release testing
of Salesforce
implementations

“

**ACCELQ is a
strong enterprise
testing choice**



HIGHEST SCORE

Among all Vendors

4.29 / 5

ACCELQ

for Salesforce

ACCELQ is a game-changer in the realm of Salesforce test automation. Its codeless approach, coupled with AI-driven capabilities, simplifies the testing process and enhances its effectiveness. For readers navigating the complexities of Salesforce testing, ACCELQ presents a compelling solution.

Salesforce Architecture	Salesforce Offerings	Diverse Technology Stack
CLASSIC	<ul style="list-style-type: none"> • Sales Cloud 	<p>Web Technologies Angular JS, React JS, Material Design, etc.</p> <p>ERP Platforms Salesforce, Pega, SAP, ServiceNow & Workday</p> <p>API Testing RESTful & SOAP based web services</p> <p>Mainframe Automation IBM, Attachmate & Hummingbird emulators</p> <p>Backend & Middleware Message Queues, Database & SSH validation</p>
LIGHTNING	<ul style="list-style-type: none"> • Services Cloud 	
LWC	<ul style="list-style-type: none"> • Financial Services Cloud • Health Cloud • Communities & More 	





ACCELQ offers **AI-powered No-Code Test Automation** and management built on a cloud-native platform. ACCELQ provides a unified platform for web, mobile, API, database, and packaged apps. Automation-first, codeless capabilities make it easy to use for testing teams without deep programming expertise.

ACCELQ allows businesses to achieve 7.5x productivity and over 72% savings with its industry-first autonomies-based automation platform. ACCELQ was named a leader in The Forrester Wave™: Continuous Automation Testing Platforms, Q4 2022.

Follow Us on



WWW.ACCELQ.COM

Dallas | San Francisco | Sydney

