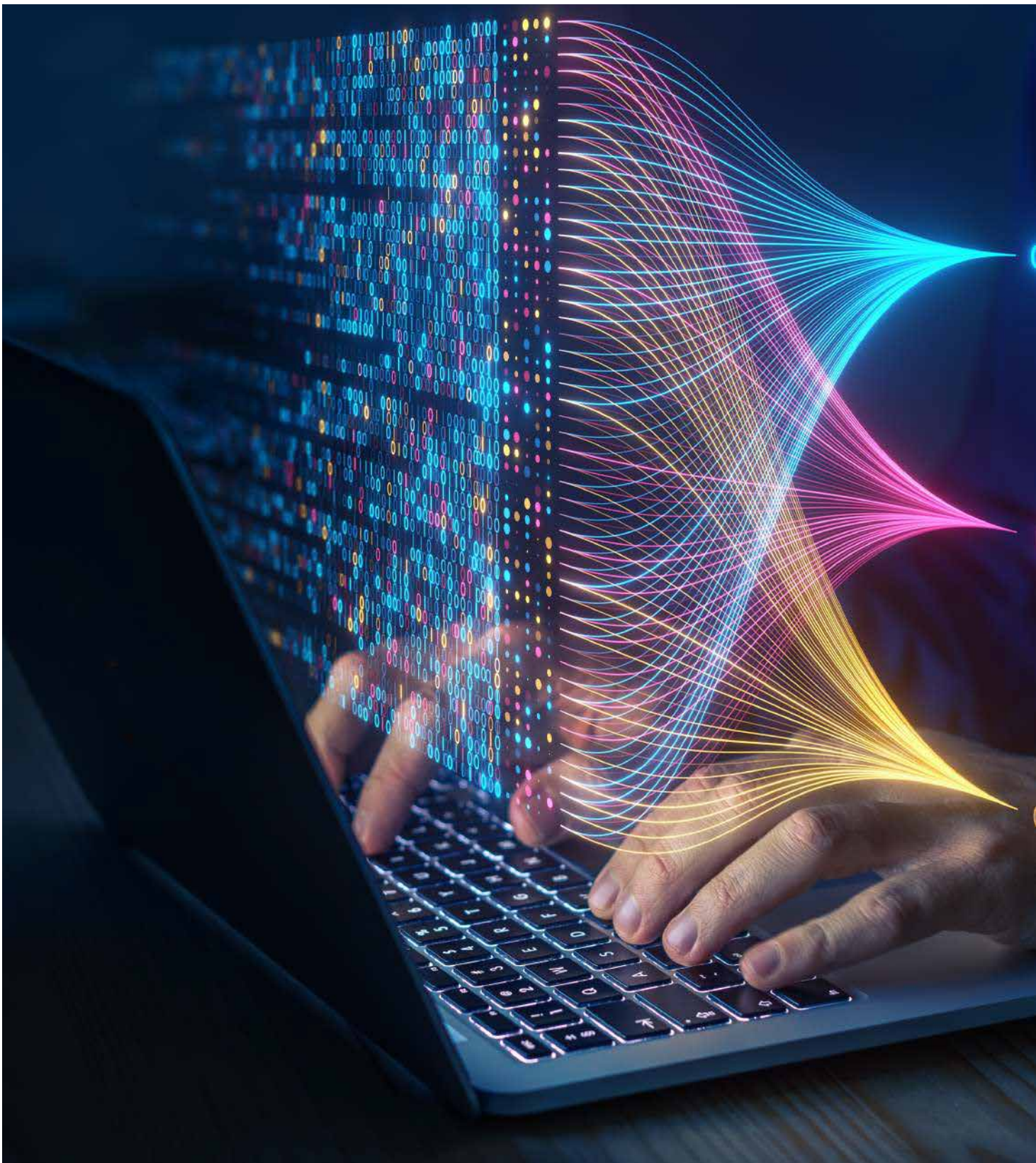


Powered By Qualitia

A Jumpstart Guide For Next-Gen Salesforce Testing



Performance of Quality Assurance deliverables like end-to-end testing and test automation remained a focused area for a long time for Salesforce. Getting it right requires a subtly refined approach, and context is crucial in choosing the right path in the present day. The dynamic nature of the platform, driven by both by Salesforce's expansion of the platform and customizations to meet business requirements, inherently involves many challenges and innovative approaches.

QualityKiosk's Salesforce Testing reference article provides a snapshot of paradigm shift in the test strategies as per technological advances. This came out after our exhaustive interaction with Salesforce professionals from IT and Business teams of the clients we delivered, to gain current expectations, challenges and best practices for this leading cloud-based application.

Table of Contents

- 1 Executive Summary
- 2 Salesforce Testing Challenges
- 3 Testing Strategies & Approaches
- 4 Testing Types & Methodologies
- 5 Salesforce Automation Testing
- 6 DevOps Integration

1. Executive Summary

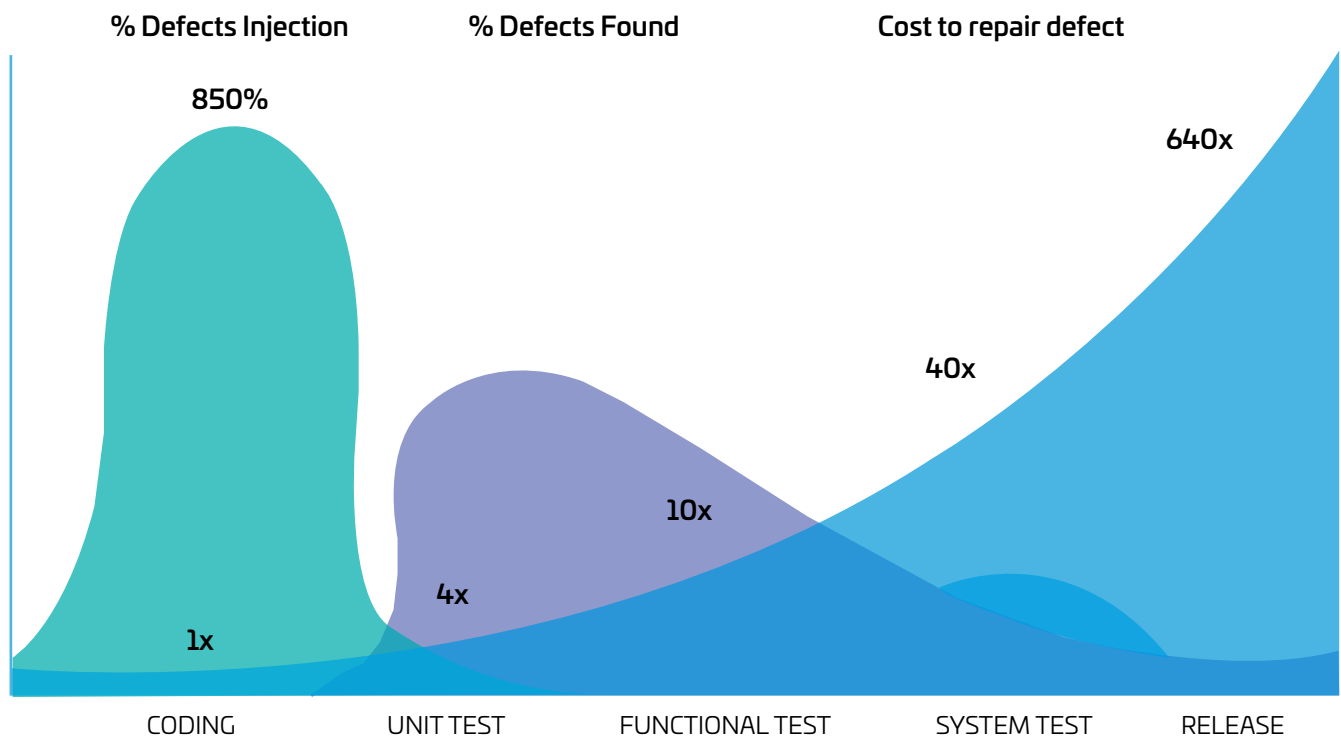
An Era of Low-Code Business Transformation

A drift towards easy-to-adapt development platforms for over a decade led to a mandate for low-code environments, starting from SMBs, due to the perpetual shortage of technical developers and the time with rapid innovations. In fact, the cardinal reason to Salesforce's success is a low-code model of development that makes it easier and faster for everyone in the process to build mission-critical apps.

Rapid innovation necessitates scalable end-to-end testing, which remains a challenge for many:

- Teams struggle with time to test scoped changes before every release
- Production issues due to inadequate testing
- Relies on manual testing for in-sprint stories

Even teams claiming to be agile often adhere to fixed timeframes like sprints — which can function like mini waterfall projects. This approach squeezes testing to the end of the process, turning into **“Water- Scrum-Fall”**



Applied Software Measurement: Global Analysis of Productivity and Quality. Caper Jones McGraw-Hill Education.2008

This manual is created to achieve four major objectives:

- Analyze current testing processes
- Measure the impact of quality
- Identify key challenges & recommendations
- Adoption of automation & value realization

2. Factors That Make Salesforce Testing Challenging

Timed Cloud Releases

Salesforce has seasonal releases for rolling out new features in the spring, summer, and winter. This is exciting for letting users take advantage of new innovations, it could mean more for testing teams.

Because platform changes can often break existing features, proper testing objectives need to be recalibrated to detect and fix issues.

Highly Customized And Deeply Integrated With Other Tools

Salesforce AppExchange offers range of integrations, which allow organizations to rapidly incorporate new functionality into the Salesforce deployments, leading to additional complexity in testing.

Adaptive Software Maintenance

Salesforce Lightning leverages many dynamic elements, such as locators or test web pages, which can be difficult to test. That is because choosing a location to test is challenging when that location is always changing, like UI/UX changes.

Table-Heavy Platform By Design

Testers may need to leverage multiple tests to validate data and ensure data communication happens as per product development.

Trained QA Experts

Though Salesforce is a user-friendly platform, it does not compliment the testing in similar way. Even some of the most accessible tools will require a certain amount of training.

Key Terminologies for Salesforce Testing

Salesforce Objects

The platform stores data in relational tables. The records in these tables contain data for the structure and user-created data. In Salesforce, objects can be classified into three main categories:

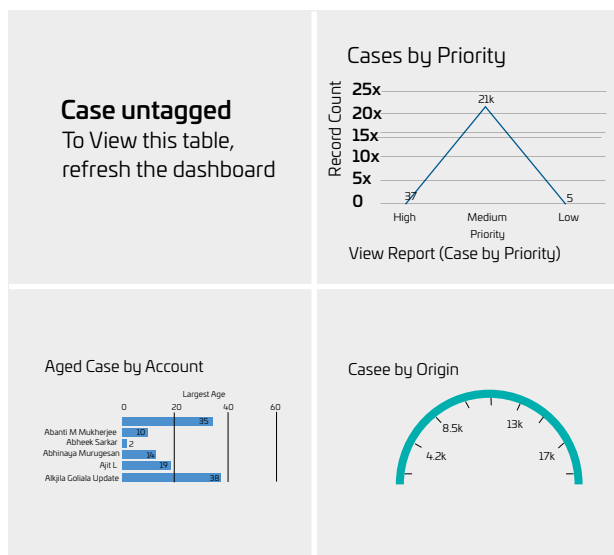
- **Standard Objects** These are the type of objects for any enterprise e.g., users, contacts, case, leads, service request etc.
- **Custom Objects** These are the objects created based on the business processes.
- **External Objects** The objects which developers create a map to the data stored for integration.

Reports

It is a list of records that meet a particular criterion or condition which gives an answer to a particular question. These records are displayed as a table that can be filtered or grouped, based on any field.

Dashboards

It is graphical notation of the reports, which shows data from source reports as visual components for charts, graphs, and tabular data.



VisualForce

visual control center that provides the ability to gather multiple languages and controls under one single tab, aiding in intuitive development and administration.

Apex

It is a programming language of Salesforce. Developers leverage it to develop custom SaaS

(Software as a Service) applications using the framework.

Governor Limits

Salesforce imposes limits on the number of operations and resources that can be used in a single transaction. These limits can impact test scenarios that involve complex workflows or data processing.

If a Salesforce Governor Limit is exceeded, the associated governor issues a runtime exception that cannot be handled. In other words, your code breaks and will not function.

Well-known verifiable governor limits with Apex programming are

- **SOQL 100 Limit** A tester should verify if apex class execute anonymous script can 'only' have 100 SELECT statements per apex transaction, check if "Anonymous Error" appear on exceeding limit.
- **SOQL 100 Limit** Fix This requires testing if the SOQL statement in a for loop, it will hit the governor limit, which is considered one of the 'venial sin' in Apex programming

DML 150 Limit

Developers in their unit testing should try to test with the use of DML statement inside for loops and check if give anonymous error. A DML operation on a collection, it only counts as one DML, this can be verification point. Additionally, AggregateResult and different collections updates the Description field on the Account object with the value of the number of Account event records.

3. Test Strategies for Salesforce

Salesforce testing in a nutshell, involves testing of enterprise deployments to ensure the product development activities are seamless & function as expected. There are three main facets to this.

As per our experiences, the Salesforce platform is utilized by companies to create bespoke, customized applications for clients' unique needs, the product needs to be tested to ensure proper performance and reliability. Through testing, testers validate whether custom development or customized application work as expected and can support the purpose behind the customizations. This includes both functional and non-functional testing.

Secondly, testing strategies are expected to deliver on all the integrations work as expected. These applications can range from 3rd party provided

solutions to proprietary applications for Order management, Transport management, Customer Service etc. All these integrations need thorough testing to ensure optimum performance and reliability.

Also, to ensure good coverage it is imperative to test the custom deployment to ensure none of the features or functionality break or get affected in any way because of the normal or ad hoc releases. This involves creating a UAT (User Acceptance Testing) testing of the scenarios as per business scenarios or product backlog as per plan.

An effective testing strategy cover experiences for both internal and external customers while

accounting for the looming risk of end-to-end changes. The right strategy will measure user risk against the costs and capabilities of a testing team to find the most efficient way to execute objectives.

- What are the testing dependencies (cross-functional)?
- How critical to the business is this scenario?
- How often could changes undermine that?
- How likely is that functionality to break?
- How long does it take to manually test?
- How likely are we to forget or run out of time to test?

4. Salesforce Testing Types

As in today's dynamic world due to intense competition in certain segments like e-commerce, media, and pharmaceuticals, the release cycles are squeezed never like before. So, during the testing process, QA teams might feel the need to pursue many different testing approaches as below:



Customer & Products

End-to-end user journeys for all possible types of Customers and Products



Data Flows

Dealing with all the data types like master & transactional data



Custom Workflow

Testing of custom workflows and business processes



Configuration

Configuration w.r.t. validation rules, formula fields, reports & notifications



Integration

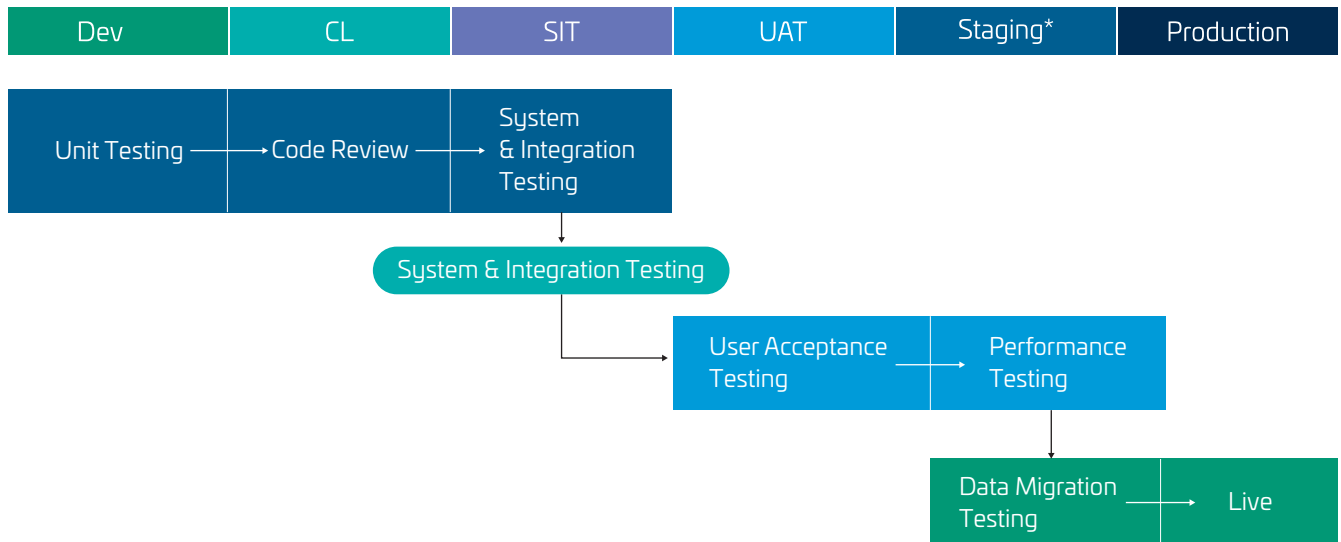
Testing of integrated interfaces and external APIs for pre-configured modules



Customization

Testing of customized Salesforce functionalities, integrated applications as per business requirement

- Indigenous Salesforce Testing covers testing options already available in the framework, Unit testing integrated into the Developer Console. This includes, different modes of data creation & storage, creation of test classes, and code coverage solutions.
- Exploratory Salesforce Testing (including data consistency validation, UI testing, user input validation, cross-browser compatibility testing, error message validation, reports and dashboard testing, flow testing, and integration testing).
- Manual Testing (functional testing, regression testing, integration testing & happy path testing).
- Automated Salesforce Testing: This has taken a center stage in Salesforce testing. A proper Salesforce automated testing can deliver reliable, scalable results while efficiently managing the resources of QA teams, though the due to dynamic nature of SFDC pages as robust test automation framework remains a challenge. Let us peek into the various testing types required across landscape.



Code-Centered Acceptance Testin

Salesforce includes complex business logic such as flows that rely on countless components under the hood. Apex code can be used to test these processes takes longer timelines.

User Interface (UI) Testing

Tools that simulate the behaviour of a user on a browser can validate if the system is working as expected.

Application Programming Interface (API) Testing

This type of test validates that APIs are responsive and return the expected information. API testing targets can provide precise verification of specific network calls.

Regression Testing

Any change to code or configuration could undermine existing functionality. These changes and updates are often seamless, but failures can remain undetected for weeks or months without regression testing.

Integration Testing - SIT (System Integration)

End-to-end functionality hinges on multiple systems working together in harmony. Integration testing makes sure business processes that involve multiple systems are firing on all cylinders.

User Acceptance Testing (UAT)

Actual end users need to verify that new functionality will meet their needs before it is deployed to production. UAT is often a final step before rolling out changes to all users.

Sanity Testing (Production)

Business users are required to check if the deployed features in production environment are working as per the requirements and as usual things are working.

Security Testing

The main goal of Security Testing and Evaluation (ST&E) is to identify the threats to the system and measure potential vulnerabilities so they cannot be exploited. It is the examination and analysis of the safeguards required to protect an information system as they have been applied in an operational environment that helps organizations steer clear of costly security breaches. Its pressing need to have a Secure Development Lifecycle (SDL) to build security into a product or application at every step in the process. (*"Get to Know Security Testing and Evaluation - Trailhead"*).

Performance Testing (PT)

It assesses system performance in terms of sensitivity, reactivity, and stability under specific workloads. Evaluating the performance and scalability of Salesforce applications to handle large user loads and concurrent transactions can be complex, especially in organizations with a high volume of users. Common Salesforce performance testing enlisted as below:

- **Measuring the Salesforce user experience**
When you measure user experience use Experienced Page Time (EPT). EPT is designed to consider the underlying architecture of Salesforce Lightning pages.
- **Server-side metrics**
EPT is measured from the client's perspective, but that's only half of the story. Salesforce is hosted by a robust infrastructure. You can gain insight into

how Salesforce is performing on the server side during a test run with Shield Event Monitoring.

There are a large number of metrics that are enabled with event monitoring. As a best practice, consider starting with the following:

- Run Time (RUN_TIME): What was the total amount of time for the request?
- CPU Time (CPU_TIME): How much time is spent in the application tier?
- Database CPU Time (DB_CPU_TIME): How much time is spent in the database tier?
- Database Total Time (DB_TOTAL_TIME): How much time does it take to make a database call?

The event types supported by event monitoring are stored in the EventLogFile standard object and their individual schemas.

Stitching Events Together

During a test run, you may need to associate multiple events together to get a full picture of what is happening. You can relate events to each other by using the fields USER_ID, SESSION_KEY, LOGIN_KEY, and REQUEST_ID. USER_ID, SESSION_KEY, and LOGIN_KEY can be used to associate events performed by a specific user or during a specific web session, REQUEST_ID is used

to associate events that are generated by a front-end request back to the Salesforce Platform.

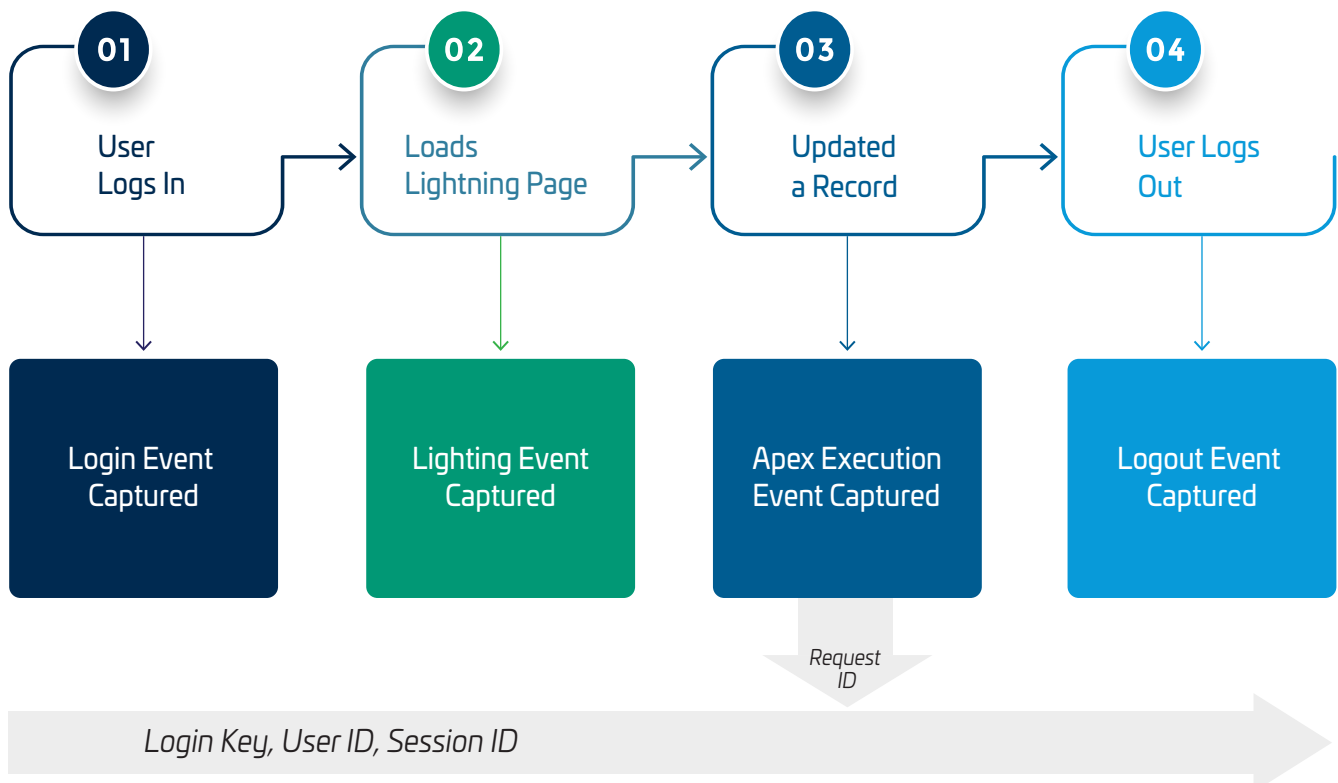
Two important performance testing metrics in Salesforce testing are throughput and response time. Throughput Systems are built to perform work for users. The number of units of work the system can process in an amount of time is called throughput. When measuring throughput, higher numbers are better.

- The number of records like Accounts, orders, service requests or cases created in a minute. The number of searches completed per hour.
- The number of product views on a community pages per day.

Response Time

Response time is the amount of time it takes for the system to process a request. You want systems to have the highest possible throughput with the lowest possible response time. Examples of response time in Salesforce are:

- The number of milliseconds to create an order.
- The number of milliseconds to load the account record page.



Mobile Testing

- **Cross-Platform Compatibility:** Tester must verify application on various operating system to check the screen size & resolution for expected user experience. different OS (Android, IOS, Symbian, Windows, etc.) but also in different versions of the same OS (like Android Nougat (7.0), Oreo (8.0), etc.).
- **Settings and Configurations Check:** The app should be compatible with all the settings & configurations of different mobile devices.
- **Test Cases Base creation w.r.t to App types:** Mobile apps are classified as web, native & hybrid. Web apps should be tested to ensure the UI is working & mobile apps have limited storage, test the memory utilization with battery usage.
- **End-to End Consistency UI Testing:** Front-end testing encapsulates the validation the UI of the app & the functionality of the mobile app graphic user interface. This test includes a drop-down menu, navigation features, and other features that are used by the end user.
- **Memory utilization & storage Test:** Test the storage should be optimum on download of multiple apps.
- **Network types of behaviour on Apps:** Performance even under weak network signals or large data transitions. Jitters, Packet loss, Network Speed.
- **Security Testing:** Source code & database, Perform input validation, Penetration test, client-side injection, check HTTPS-SSL/TLS security layer, local data storage.
- **Geolocation Testing:** Localization test intends to validate, globally operating software to provide linguistic and cultural relevance. It is a technique to verify software behaviour, accuracy, and suitability for specific locations and regions.

5. Salesforce Test Automation

Automated testing enables fast and reliable test scripts via low-code platforms are the effective way to slash the cost of investment and reduce business risk. Repetitive tasks can be automated - freeing up users to focus on core work; exploratory testing, understanding the scope of testing from PO/Developers etc., test case design to ensure complete coverage, analysing test plans, communicating with users and more.

Based on our delivery experiences, testing remains the biggest bottleneck in the process of digital transformation. Clients across industries are still using traditional testing methods such as manual testing or script-based approaches. As a result, teams lack the time to test changes before deployments and the release quality suffers. Test automation can relieve some of the pressure on the quality assurance (QA) team.

However, we have also observed a growing shift from manual testing to automated testing, which has significant outcomes for teams leveraging the potential of automation:

- Fewer production failures after each release
- More frequent release (bi-monthly/weekly)
- Better completion of testing for each release
- Lower TCO (Total Costs of Ownership)

Adopting the low-code approach accelerates development. With Salesforce managing the underlying infrastructure, faster software cycles result in more changes and updates, potentially leading to unexpected surprises.

The development of mobile apps is expected to be swift and reliable, so testing automation is at the core of the apps. Testing strategy. It ensures that the mobile application covers customer expectations and business objectives. But the testing has its own challenges on the cost optimization along with expectations to support various types of mobile devices, and operating systems like Android, iOS, and Windows.

To address mobile testing and its recurring activities, because of multiple OS upgrades, the launch of new device models, new releases to remain competitive on mobile features, etc., automation using effective tools can be leveraged with the following advantages.

Codeless test
creation

User-friendly
UI for testing
activities

Support all
actions on screens,
i.e., swipe, pinch,
rotate, tap

Cross-platform
development support
(same code for
Android & iOS device)

Supports multiple
languages

Does not require
source code access

Supports Android and
iOS-based apps

Incredible
customer support

6. DevOps Integration for Regression Automation

DevOps for SAAS (Software as a Service) based platform like Salesforce has emerged as one of the hottest trends in the low-code-No-code ecosystem.

Continuous updating & Integration (CI) of the test suites for regression testing becomes the trends to catch-up with the watertight timelines.

Automated testing runs minimizes this risk by providing a layer of protection — allowing functionality to be checked and rechecked many times with minimal human intervention on daily basis. Creating an automated test requires initial effort, but the cost of re-running each test is worth having it. This makes test automation an ideal solution for high-velocity teams that need fast and consistent testing to prevent backward regression as they build new capabilities

About QualityKiosk Technologies

QualityKiosk Technologies is one of the world's largest independent Quality Assurance (QA) providers and digital transformation enablers, helping companies build and manage applications for optimal performance and user experience. The organization has been featured in renowned global advisory firms' reports, including Forrester, Gartner, and The Everest Group, for its innovative, IP-led quality assurance solutions and the positive impact it has created for its clients in the fast-changing digital landscape.

Contact us at letsconnect@qualitykiosk.com for a complimentary consultation on QA transformation and take the first step to future-proofing your digital transformation.