



- Vijayanathan Naganathan, Sreesankar Sankarayya

Abstract

In today's volatile economy, organizations can meet business demands of faster time to market and scalable costs by leveraging SaaS, thus helping them focus on their core business. This inherent advantage has led to the large scale interest in possible SaaS adoption. However, the expectations from organisations that are subscribing to SaaS are very high, since they perceive SaaS to be a complete and customized solution to their business needs. Given this, in order to ensure that organisations can reap all the benefits of SaaS, and meet the expected business demands, comprehensive testing is essential.

In the previous paper of this series¹ we had discussed and understood what SaaS testing comprised of and the typical challenges encountered while validating SaaS based applications. This paper takes you through as to how we can address all the challenges associated with SaaS testing and ensure that we are able to extract maximum benefits from our SaaS based implementations.

Overcoming challenges in SaaS testing

The growing demand of SaaS based applications has increased the demand for SaaS Testing as well. However, there are multiple challenges that are encountered in the entire course of SaaS Testing. In the following table (Refer Table 1), we look at how we can overcome these challenges, leading to the enablement of end-to-end testing for SaaS based applications.

Challenges	Overcoming These Challenges
Testing frequent SaaS upgrades – Short notice period (1-2 weeks) for a QA notification to validate the application	The use of automation tools for building regression suites brings in business value and helps quickly validate the impact of upgrades.
Business knowledge for effective testing of configurable and non-configurable components	Gain comprehensive and competent knowledge on the configurable and non- configurable components of SaaS applications
	Any non-configurable upgrade/change to the application will need to be assessed thoroughly since this will have an impact on all SaaS subscribers
	Though the configurable upgrade/change would not impact every client, it is advised to validate the impact of these changes as well
Validating interface compatibility	The backward compatibility of a SaaS interface needs to be validated to ensure that the organizations do not have to make any changes at their end, and can continue using SaaS applications as before.
Compliance with government regulations and other standards	Devise a unique comprehensive testing strategy for compliance with standards like PCI and government regulations.
Data security and privacy	Validation of strong encryptions is needed to ensure data security
	Data security and privacy would need to be thoroughly validated amongst multiple tenant scenarios to ensure that there are no loop holes
Testing access controls, multi-privileges for security	• Perform access control and multi-privilege tests with users that have varied roles, different privileges and are executing unique activities (simulating real life usage scenarios).
Data integration - inbound & outbound	 Test data transfers between an organisation's network and SaaS applications. Also, measure, compare and validate the performance of data migrations between SaaS applications and an organisation's network.
Simulating live upgrade testing	Live upgrade tests should be carried out in cloud based pre-production environments.
	Use automation tools to simulate the scenario of multiple concurrent users logged on to a current SaaS version.
	Conduct live upgrades in cloud based environments.
	Use automation tools to validate the accuracy of the upgrade.
Optimization of testing that is common to the impacted core and non-core areas of SaaS when getting customized	Create a test strategy to test the core product of SaaS.
	Create a standard suite of automated test cases to validate the core SaaS product.
	Create a map/grid of the core and the non-core areas of the SaaS application that are most likely to be impacted during customization.
	Run a regression suite selecting the tests associated with the impacted areas.
Data migration from the existing system to SaaS application	• Identify the different data sources in the existing system that need to be migrated to the SaaS application.
	Select tools that will help in the data migration and in the post migration validation.
Frequent releases of feature rich SaaS applications increases the time taken for testing, owing to the significant number of pages to be covered	Create an automated test library for SaaS applications that help reduce the associated testing effort that comes with each frequent release.
Rapid addition of new features to the core SaaS product to meet new customer demands and to stay competitive. However, every change is a potential security bug/ performance issue.	Formulate a comprehensive strategy for testing the SaaS applications with test tools that cover functional, performance and security requirements.
	Maintain a test repository of results, performance benchmarks and access privilege grids, which would facilitate faster validation.
	Execute comprehensive tests with automated tools that cover the functional and non-functional requirements.
	Conduct a continual impact analysis of requirements and regularly update the test library to help minimize risks.

Table 1: Overcoming Challenges Associated with SaaS Testing

The focus of SaaS testing

As illustrated in Figure 1 (The SaaS Testing Focus), SaaS testing needs to encompass a wide range of functional and non-functional requirements (performance, security and compliance) including the application, cloud characteristics and SaaS attributes.

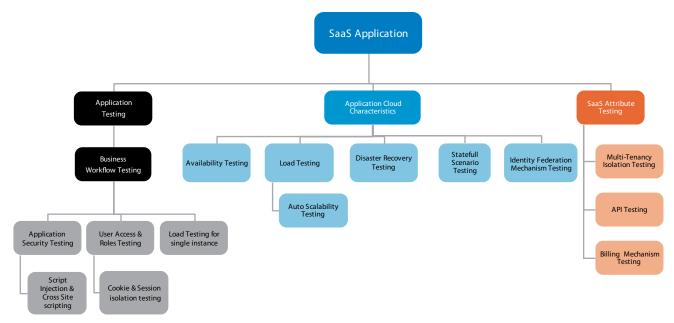


Figure 1: The SaaS Testing Focus

The functional requirements of SaaS applications can be addressed by testing the line items mentioned under business testing in figure 2 (The broad classification of the different types of SaaS testing) and the non-functional requirements can be addressed by testing the line items mentioned under security and performance testing. Moving beyond functional and non-functional requirements, emphasis also needs to be laid on testing the operational aspects of the SaaS application. These are taken care of by testing the line items listed under compatibility, live and SaaS attribute testing, as shown in figure 2.

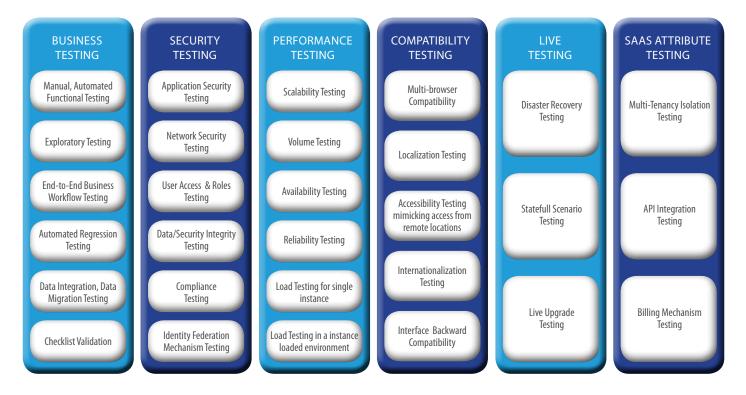


Figure 2: The broad classification of the different types of SaaS testing

The Implementation of SaaS Testing

Now, let's take a look at the SaaS testing process itself. SaaS testing begins with assessing the functional and non-functional requirements for the SaaS application, including business, operational and non-functional needs. Once this is done, the focus then moves into understanding the usage pattern of the application. This particular set factors in the variations due to geographies, peak periods and network latencies across regions.

A test plan would need to be developed to include all components of the SaaS application. The plan would also have details on how these components would be tested and the resources needed to carry out the same. Once the test plan is approved, the QA team would prepare test cases, test suites and eventually get the test data ready. The QA environment is then validated for its preparedness for SaaS Testing. After the assessments confirm the preparedness, test data is populated in the QA environment through data migration from the existing system. Then the test team focuses on the automated test suite generation for functional and non-functional validations. This would be followed by test execution, reporting, publishing and finally culminate with the issuance of the SaaS readiness certification. See figure 3 (The SaaS Testing Process) to get details on all steps and processes required for ensuring a systematic and successful SaaS Testing.



Figure 3: The SaaS Testing Process

The Benefits of SaaS testing

There are multiple benefits that SaaS testing delivers to organisations.

- Reduces effort required and go-to-market time associated in procurement, upgrades, renewals, contracts, maintenance and deployment.
- Lowers costs associated with test tools, test environments, maintenance and upgrades.
- Helps focus on the SaaS application configuration rather than on provisioning for the application and associated infrastructure requirements.
- Significantly reduces CAPEX associated with setting up of environment for SaaS application, helping convert the same into OPEX
- Reduces shelf ware risk of SaaS application and testing tools associated with the validation of the application.
- Testing costs are reduced by almost one third as the need to test client server installations, multi-platform backend support, multiple versions of upgrades and backward compatibility is completely eliminated.
- Using SaaS testing tools are not system or machine dependent.
 For example, any local machine connected to a cloud network can be used for performance testing of the SaaS application.
 This helps save effort and overhead expenses associated with the installation, configuration and maintenance of additional machines for enabling SaaS testing tools.

Conclusion

SaaS testing focuses on ensuring high quality across the application, its cloud characteristics and SaaS attributes. It also includes testing for security, privacy, accessibility and standards compliance as well. A thorough understanding of the SaaS application, the customer specific implementation, components that are configurable and nonconfigurable and how any change or upgrade would impact the application is absolutely needed to ensure a successful SaaS application testing. The automated validation of the functional and non-functional requirements of the SaaS application helps shorten the release cycle of frequent SaaS application upgrades and releases. The data integration/migration pertaining to SaaS applications would also need thorough validation. The key to successful SaaS testing is putting together the right test strategy, automating the tests for functional and non-functional requirements and leveraging best practices that would help maximise the investments in SaaS and in turn help the organization achieve the intended business outcome.

References

1. The Challenges Associated with SaaS Testing, Infosys, October 2011

About the Authors

Vijayanathan Naganathan (Vijayanathan_n@infosys.com) is a Senior Technology Architect with the Independent Validation group at Infosys. With 13 years of industry experience, he currently leads the Cloud and Service Virtualization service offerings for QA. His current work includes helping customers adopt cloud for QA environments, defining strategies and executing them for cloud based application validation.

Vijay blogs at http://www.infosysblogs.com/testing-services/

Sreesankar Sankarayya (Sreesankar_s@infosys.com) is a Technology Lead with the CLOUD group at Infosys. With 7 years of industrial experience, he currently works on the Microsoft Azure Platform and focused on devising and proving methods to make Java/J2EE applications compatible and portable to the Azure PaaS.

