QATestLab
quality is a rule

# Software as a Service (SaaS):

## Survey, Specifics, and Testing

# Table of Contents

Click the section to jump ahead



# Introduction

Software as a Service (SaaS) is turning into the most popular software delivery model in today's world. Currently, the absolute majority of people and companies use 1 to several SaaS solutions on a daily basis. Even though sometimes they may not even know that the application they are using is SaaS.

So, what is Software as a Service (SaaS), and what makes it so unique? What are its pros and cons? And how does one test SaaS solutions? In this whitepaper we answer these questions and many others.

The purpose of this work is to explain all the aspects of SaaS and its testing. It contains extensive information on the essence and specifics of Software as a Service, as well as practical recommendations, actionable advice, and useful tips on how to test SaaS solutions, including:

- a comprehensive checklist of the Quality Assurance (QA) activities for SaaS applications
- a detailed list of challenges of SaaS software testing and ways to deal with them
- extensive information on the best practices of testing SaaS solutions to get the best results in the most efficient way

# 1 The Software as a Service (SaaS) model

This part of the whitepaper gives an overview of what Software as a Service (SaaS) is, explains how it's different from other software delivery models, and describes SaaS pros and cons.

In the first chapter, we provide general information about the SaaS model to explain its essence and key characteristics.

Then, we delve more into the details, describing the specifics of Software as a Service. This chapter explains the differences between the SaaS model and the traditional model of software delivery. Understanding what makes SaaS unique is essential when deciding to use, provide, or test SaaS solutions.

After that, the advantages of the SaaS software for its users and vendors are described.

Finally, we address the challenges and risks associated with the SaaS model for SaaS providers and customers. We hope that the information about the pros and cons of SaaS will help the readers to get the full picture of this software delivery model and decide if it is the right choice for them.

# 1.1 What is SaaS

**Software as a service (SaaS) is** a model of delivering a ready-made software solution to end-users over the Internet on a subscription basis.

Unlike the traditional software delivery model, SaaS doesn't imply purchasing a one-time license to get access to the product or the necessity to install the solution on the devices, on which it will be used. Instead, regular fees are paid for the usage of SaaS software that can be accessed from anywhere and through any device via the Internet. In most cases, SaaS solutions are used through web browsers. But mobile applications are also becoming an increasingly popular way to access SaaS products.

One of the important characteristics of SaaS software is that its vendors are completely responsible for its maintenance, updates, upgrades, and fixes of any issues that arise. SaaS clients do not need to concern themselves with any of these tasks. As soon as they subscribe for a SaaS product, they can start using the ready-made product with no additional difficulties. This is why SaaS software is sometimes called on-demand software.

SaaS is also one of the major categories of cloud computing. SaaS solutions are hosted in the cloud. In case of larger companies who provide SaaS applications, they will also own the cloud where the applications are hosted. Smaller SaaS vendors purchase hosting for their solutions from third-party cloud providers.

SaaS solutions are diverse and extremely popular nowadays, so there are few people who have never used them. Email services (such as Gmail, Yahoo, Mailchimp, etc.), popular cloud programs (Google Docs, Microsoft 365, Zoom, etc.), online subscription-based entertainment (for example, Netflix), and cloud business solutions (Salesforce, Slack, DocuSign, etc.) are all examples of SaaS software.

# 1.2 SaaS specifics

## MULTI-TENANCY

Multi-tenant approach is typical of SaaS solutions. In it, a single cloud instance of the SaaS software serves multiple tenants or subscribing customers of the SaaS application. The version of the SaaS software is the same for all the users, and all the updates, upgrades, and other changes are rolled out to all the SaaS customers at once. Despite working on a single cloud instance of the SaaS software, different customers' accounts and information are still separated and protected so that no data leakage should occur.

## ACCESS OVER THE INTERNET

With SaaS applications, there is no need to install software on any devices where it will be used. The distinctive feature of SaaS solutions is that they are accessed over the Internet. They are ready-made applications that can be reached and used online at once through any device with Internet access a customer has at hand.

### SUBSCRIPTION-BASED PAYMENT

SaaS users don't purchase the software through buying a license but rather rent this software by paying regular fees. So the SaaS model implies recurring but more affordable payments. The cost of the subscription already includes all the expenses for maintenance, upgrades, and bug fixes. It increases if SaaS customers decide to order more functionality or resources.

### RETAINING EXISTING CUSTOMERS AS THE MAIN BUSINESS PRIORITY

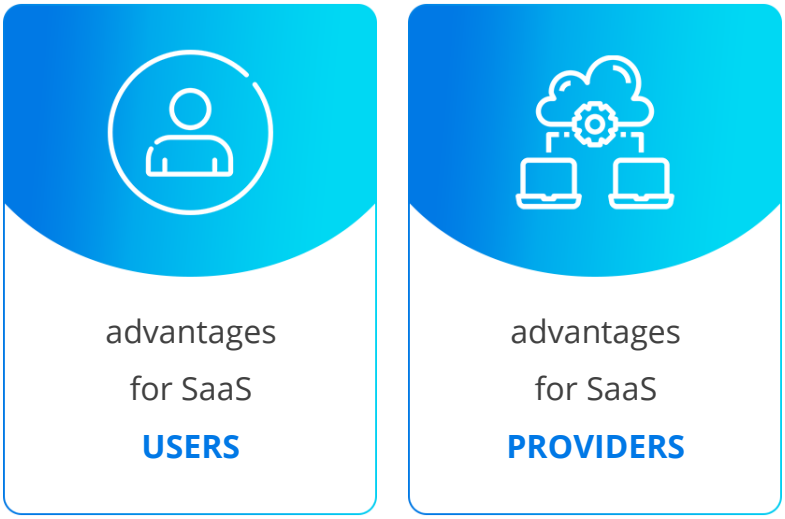
In the traditional license-based software delivery model, companies receive the largest income when a new customer buys the license. Purchasing further upgrades of the software is cheaper. So in the traditional model, attracting new customers is the primary source of income. In the SaaS model, all the subscription payments are equally distributed. Long-time clients are also more likely to upgrade their subscription plan and order additional resources or functionality. So in the SaaS model, retaining existing clients is the main business priority. And it outweighs attracting new customers.

# 1.3 SaaS advantages

The current popularity of the SaaS model can be explained by multiple unique advantages it offers both to the customers (be it individual users or businesses) and to the SaaS vendors.

In this chapter, we will delve into the details of specific advantages the SaaS model provides.

For the readers' convenience, the benefits were divided into two large groups:




We hope that the chapter will explain why so many people nowadays choose the SaaS model. Another purpose of this section is to help the readers decide if Software as a Service is right for them.

# 1.3.1 Advantages for SaaS users

### FAST DEPLOYMENT

Users can start benefiting from a SaaS solution as soon as they order it. There is no need to wait for the application to be completed, as with on-premise software development. And there is no need to spend time and effort on installing the application on every device, as with traditional software. The only thing needed to start using this type of software is Internet access.

### ACCESSIBILITY

SaaS solutions don't require any particular hardware specifications. They can be accessed through any device online from anywhere in the world. SaaS applications can also be used by multiple users at once. It makes this type of software especially useful for companies with a large percentage of remote staff and distributed offices.

### VENDOR-SUPPLIED SUPPORT, MAINTENANCE, and UPDATES

A SaaS provider is responsible for the solution maintenance and updates that are automatically applied to all the platform's users. An easy-to-reach support is also a standard in this business model. Usually, the vendor performs data backups as well. This makes SaaS more convenient and hassle-free than traditional software. It also reduces the burden and dependency on the customer's internal IT staff.

## COST EFFICIENCY

In the SaaS model, users pay recurring fees for a subscription instead of making a single purchase of an expensive license. This makes SaaS more affordable and helps to plan one's budget more easily. All updates, maintenance, and support are already included in the fee. So, SaaS clients can save money on hardware, their own data centers, infrastructure, IT staff, and purchasing software updates. The cost also normally depends on how much a client uses the solution or how many users use it, which prevents overpaying for unneeded services or resources. And many SaaS companies offer a free trial period so that the customers can see if this application is what they need. Overall, SaaS solutions are considered cheaper than traditional software.

## SCALABILITY

SaaS offers great consumption flexibility. Users can scale up any time they need more services, resources, or functionality from a SaaS solution. Similarly, they can scale down and save money when they don't require some of the offers their subscription plan implies.

## INTEGRATION WITH OTHER SOFTWARE

SaaS solutions typically allow easy integration with other applications. This is especially so when the users want to integrate a SaaS application with the software of the same provider. But the SaaS model overall promotes interconnectivity and recognizes the ways it can bring more benefits to users. So integration with third-party applications is also usually easier in SaaS than in traditional licensed software that tends to be more isolated.

# 1.3.2 Advantages for SaaS providers

### STABLE INCOME

Recurring subscription payments - instead of a one-time license purchase of the traditional software business model - allow SaaS providers to receive steady revenue from month to month. This makes it easier for such companies to create a long-term business strategy and to grow and develop more efficiently.

### PROTECTION FROM PIRACY

SaaS users don't get access to the source code of the software, instead using the product only through a web browser or a mobile application. This makes creating illegal copies of the SaaS software much more complicated than with traditional software. Thus, SaaS providers are better protected from software piracy.

### EASIER UPDATES

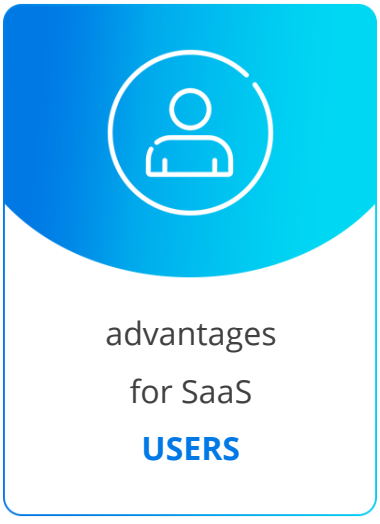
Because of the multi-tenant architecture of SaaS software, the vendors can easier perform maintenance, manage updates, and implement changes in their application. Instead of doing it in multiple instances for separate users, the providers can perform the needed changes for all the users at once, in one instance. As SaaS software is updated for all the customers at once, the providers also don't need to continue supporting multiple older versions of the software. All of it lessens the burden on the IT staff of a SaaS company and helps to decrease the expenses on personnel.

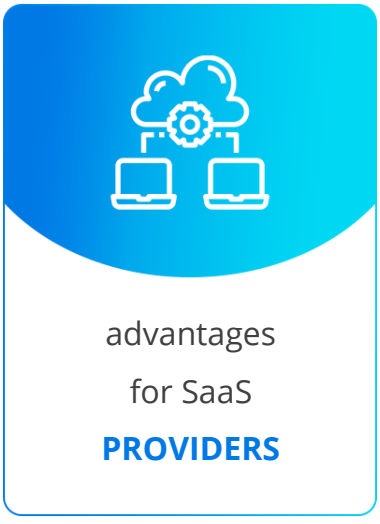
# 1.4 SaaS challenges and risks

Despite its many advantages and the multiple unique benefits it can bring, the SaaS model has a number of downsides as well.

In this chapter, we list and describe the challenges and risks associated with Software as a Service.

As with the SaaS advantages, its possible negative or concerning aspects are also divided into two large groups:




Our aim is to provide the readers with a balanced overview of SaaS, addressing both its pros and cons. By describing the challenges and risks of the Software as a Service model, we also want to help the people interested in it make more informed decisions.

# 1.4.1 Challenges and risks for SaaS users

### SECURITY

Security is one of the major concerns for SaaS applications. As these solutions are accessed via the Internet and have a shared multi-tenant infrastructure, they can be more vulnerable to security and data privacy issues. SaaS applications are also very popular and have many individual and corporate users, which makes them an attractive target for cyber attacks. This is why these solutions can experience hacking, phishing, data theft, distributed denial-of-service (DDoS), and SQL injection attacks, etc. and need to have really strong security protection measures in place.

### LIMITED CONTROL

With a SaaS solution, the user completely relies on the provider for maintenance, updates, support, and overall ensuring the proper functioning of the product. If anything goes wrong on the provider's side, such as a service disruption or a security breach, the customer has no control over it and has to depend on the provider to fix it, while suffering the issue's consequences. The vendor can also enforce certain software changes that create difficulties for the user. For example, SaaS solution updates are rolled out to all users. Some of these users may prefer to continue using the older version of the software but do not have such an opportunity. Others may be forced to spend extra resources and time on training their employees to adjust to the software changes. Hence, a lack of control over a SaaS solution may be a serious disadvantage for some users.

Click the section to jump ahead

Introduction
1. The Software as a Service (SaaS) model
1.1. What is SaaS
1.2. SaaS specifics
1.3. SaaS advantages
1.3.1. Advantages for SaaS users
1.3.2. Advantages for SaaS providers
1.4. SaaS challenges and risks
1.4.1. Challenges and risks for SaaS users
1.4.2. Challenges and risks for SaaS providers
2. Testing of SaaS solutions
2.1. Specifics of testing SaaS software
2.2. Checklist for SaaS testing
2.2.1. Functional testing
2.2.2. Security testing
2.2.3. Performance testing
2.2.4. Compatibility and Interoperability testing
2.2.5. API testing
2.2.6. Usability testing
2.2.7. Localization testing
2.3. Challenges of SaaS testing
2.4. Best practices for SaaS testing
Conclusion
About QATestLab
Contacts

## ISSUES WITH MIGRATION

The process of migrating the user's data from an on-premise application to a SaaS solution and especially from one SaaS provider to another can be challenging. The issue has to do with the large amounts of data that need to be transferred and with ensuring that everything works properly after the transfer. Another difficulty is that SaaS vendors may sometimes use proprietary technologies that are incompatible with the solutions of other providers. This might cause vendor lock-in, a situation when a user cannot change the service provider because of the significant difficulties associated with the change.

## INSUFFICIENT CUSTOMIZATION

While most SaaS solutions offer customization options to various degrees, it may not be sufficient for some users. This type of software is created to meet the general demand of the targeted audience. So it is possible that the SaaS solution the potential user is interested in does not offer the customization opportunities required for the specific needs of this user. In this case, individually developed software can be a better option.

## INTERNET CONNECTION

As SaaS solutions are accessed online, any issues with the Internet can have an instant negative impact on the users. When there is no Internet connection, SaaS software cannot be used. And when the Internet speed is slow, there will be issues in the performance of SaaS solutions as well. They can work slowly, or there may be other disruptions like some options not working properly because of the insufficient speed of the Internet connection.

# 1.4.2 Challenges and risks for SaaS providers

### ENSURING SECURITY

Just like possible security issues are one of the biggest concerns of SaaS users, preventing such problems and ensuring the security of a SaaS application is one of the main challenges for SaaS providers. To deal with these issues, vendors need to follow the best security practices and always have security of their application as their high priority.

### SCALABLE CONSUMPTION OF CLOUD RESOURCES

Easily scaling the number of users and allocated resources up and down is one of the key characteristics and major advantages of SaaS applications. In order to do that, providers usually use auto-scaling cloud tools. But for these tools to work correctly and efficiently optimize the resource usage, they need to be properly configured and monitored for healthy performance. SaaS providers need to pay attention to this task.

Click the section to jump ahead

Introduction
1. The Software as a Service (SaaS) model
1.1. What is SaaS
1.2. SaaS specifics
1.3. SaaS advantages
1.3.1. Advantages for SaaS users
1.3.2. Advantages for SaaS providers
1.4. SaaS challenges and risks
1.4.1. Challenges and risks for SaaS users
1.4.2. Challenges and risks for SaaS providers
2. Testing of SaaS solutions
2.1. Specifics of testing SaaS software
2.2. Checklist for SaaS testing
2.2.1. Functional testing
2.2.2. Security testing
2.2.3. Performance testing
2.2.4. Compatibility and Interoperability testing
2.2.5. API testing
2.2.6. Usability testing
2.2.7. Localization testing
2.3. Challenges of SaaS testing
2.4. Best practices for SaaS testing
Conclusion
About QATestLab
Contacts

## QUICK UPGRADES, UPDATES, AND BUG FIXES

With SaaS applications, users expect that any issue that arises is fixed as soon as possible, preferably no longer than in a couple of hours. It is the responsibility of a SaaS provider to organize the work and manage the application in such a way that it is possible. If SaaS vendors fail to ensure quick enough bug fixes, updates, and upgrades, it will be difficult for them to retain customers.

## CUSTOMIZATION

A SaaS solution is developed to meet the general needs of a large group of individual customers and companies, but its users will also expect a certain level of customization available. Finding a fine balance between customization opportunities and not losing sight of the application's main purpose and its users' primary needs can be a challenge for SaaS providers. Once customization becomes available for SaaS users, vendors also need to make sure that no changes or updates in the application have a negative effect on the customization opportunities and the users' individual configuration settings.

# 2

# Testing of SaaS solutions

This part of the whitepaper is dedicated to all the main aspects of testing SaaS solutions. Quality assurance of this type of software has many unique characteristics and specifics one needs to know and understand to make the process efficient. We strive to describe them all in a succinct and practical manner.

First, we will address the ways in which testing SaaS solutions differs from testing traditional software. The chapter dedicated to this question will give the readers an overview of the specifics of SaaS testing so that they can understand the process better and be prepared to deal with its distinctive aspects.

Then, we will delve into a more practical side of things by providing a comprehensive checklist for testing a SaaS solution. It will cover all the essential types of testing SaaS software so that the readers can easily make sure they checked all the main areas of the solution during QA.

After that, we will describe the challenges of SaaS testing. The chapter addressing this question will explain what can make quality assurance of a SaaS solution difficult and what one needs to be particularly concerned with when testing such software. It will help the readers to be prepared for the challenges and know what to pay special attention to during QA.

Finally, we will provide information on the best practices of SaaS testing. It will help the readers to effectively deal with possible challenges of performing QA of a SaaS solution that were described before. This chapter will contain much actionable advice on how to optimize SaaS testing and achieve the best results with it.

# 2.1 Specifics of testing SaaS software

- As SaaS software is released and upgraded faster than traditional software, this puts more emphasis and pressure on the Quality Assurance for SaaS solutions. There is a larger requirement for the speed and efficiency of software testing activities than with traditional software.
- To properly test all configurable and non-configurable elements of a SaaS application, a QA engineer must have extensive knowledge of cloud applications testing.
- The users of SaaS applications often actively participate in the testing process, specifically beta testing and experimental testing of new features. It allows SaaS customers to ensure the fullest usage of the benefits of a SaaS solution.
- Setting a proper testing environment becomes particularly important when testing a SaaS application. The production environment cannot be used for testing because it can disrupt the SaaS solution performance and negatively impact the existing users.
- SaaS testing can include three main components:
  1) testing of the application itself (its functionality, security, compatibility, workflows, etc.)
  2) network testing (checking the proper transfer of data, network bandwidths and protocols, etc.)
  3) infrastructure testing (disaster recovery, secure connection, regulatory compliances, backups, storage policies, etc.)

Testing SaaS applications is often considered more difficult than testing traditional software.

# 2.2 Checklist for SaaS testing

When performing Quality Assurance activities, a checklist can be of much help to make sure all the necessary tests were performed and all the important aspects and areas of the solution under test were covered.

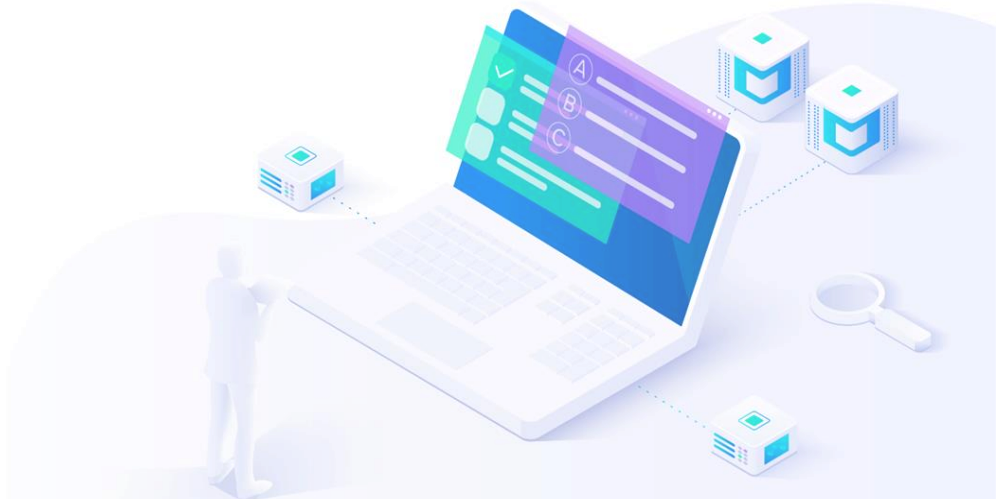
In this chapter, we will provide a checklist including the main types of testing essential for SaaS QA:

- **functional** testing
- **compatibility** and **interoperability** testing
- **security** testing
- **usability** testing
- **performance** testing
- **localization** testing
- **API** testing

Our goal is to make the testing process easier for the readers by giving them actionable advice and tips on what needs to be checked in every mentioned testing type during the QA of a SaaS solution.

# 2.2.1 Functional testing

- Test if all the functionality is working according to the requirements.
- Perform the solution's build verification.
- Conduct system testing.
- Check the components' integration in the application.
- Assess the configuration and customization opportunities of the solution and their work.
- Perform thorough manual testing, keeping the end user in mind.
- Conduct exploratory testing.
- Test the solution for end-to-end scenarios.
- Perform regression testing after each change in the solution as well as after migration and integration processes.
- Check the SaaS application in an environment with multiple users where different users perform different activities.



# 2.2.2 Security testing

- Test the security of a SaaS application for typical threats such as SQL, LDAP, SMTP injections, cross site scripting, HTTP header injections, etc.
- Evaluate network security for secured connection, controlled access to servers and sites, backup and storage policies, etc.
- Test for possible cyber attacks.
- Check the various roles and privilege levels in a multi-tenant environment.
- Assess all the upgrades for security issues. Make sure all security patches are present.
- Test the integrity and security of data among multiple tenants.
- Identify the scenarios in which the SaaS application could be vulnerable.
- Check the SaaS solution for the Payment Card Industry Data Security Standard (PCI) compliance.
- Assess the possibilities of data leakage, issues with SSL configuration and cookie settings, etc.
- Collect logs about suspicious requests and security warnings.
- Test for attempts to tamper with the licensing, gaining unauthorized data access, and editing configuration files or registry entries.
- Check for such authentication vulnerabilities as role manipulation, user enumeration, path traversal, etc.
- Test for information gathering vulnerabilities like application entry points, application discovery, etc.
- Assess security during integration and migration processes.

# 2.2.3 Performance testing

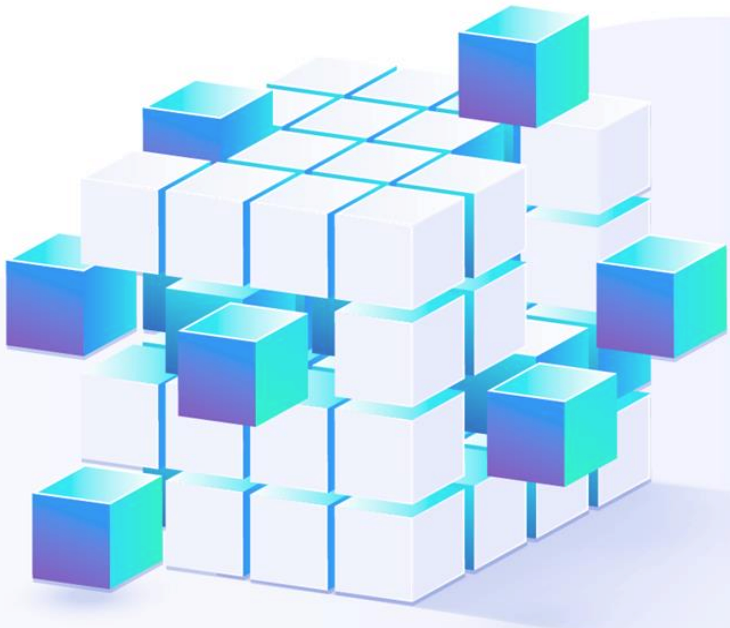
- Test if the SaaS application can handle large data volumes.
  Verify if the expected number of simultaneous users can be properly supported.
- Assess the application's response to normal, continuous, and peak loads in multiple environments.
- Check the solution's CPU usage, memory consumption and response time based on the requirements.
- Verify whether there is global access to the application and if the response time is acceptable.
- Check if the application is available 24/7.
- Test if the solution can scale up and down as per the requirements.
- Check the application for disaster recovery, data restoration, and rolling back to the last stable version.
- Test the performance of the solution in case of live updates and deployments.

# 2.2.4 Compatibility and Interoperability testing

- Conduct Interoperability & Compatibility Testing to decrease the number of errors during implementation.
- Check the SaaS application for cross-browser and cross-platform compatibility.

# 2.2.5 API testing

- Check if the SaaS solution can be smoothly integrated with other third-party applications and enterprise software of the SaaS users.
- Test if input and output parameters of APIs are correct. Check the response time and if it is according to the expectations.
- Verify if the changes to APIs cause any malfunctioning of the SaaS solution.

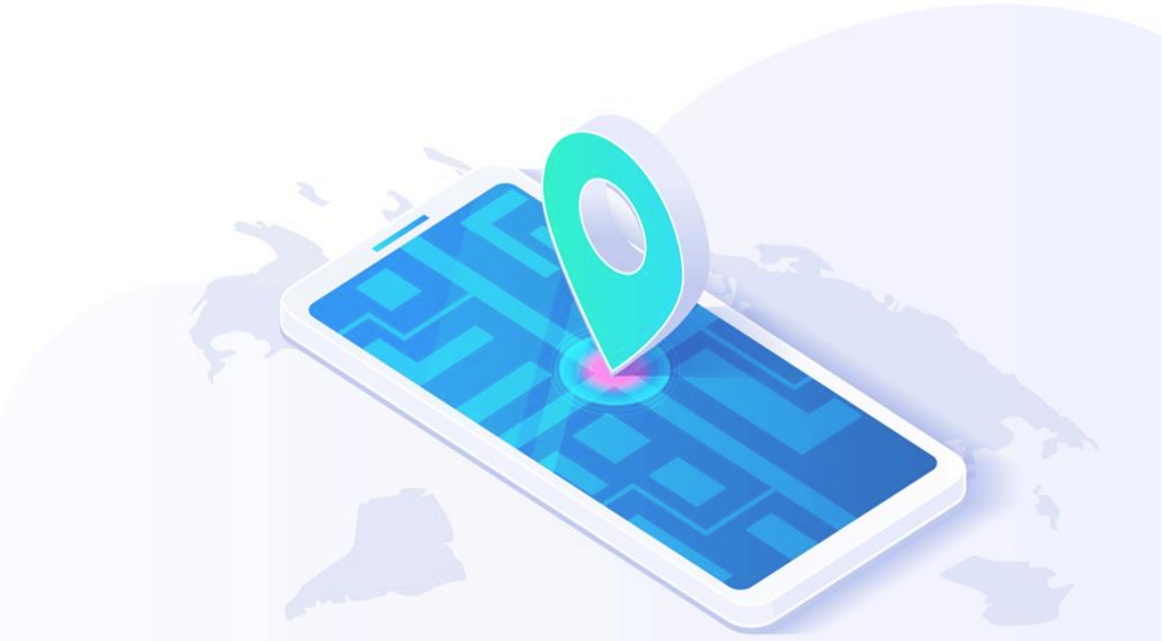


# 2.2.6 Usability testing

- Check if the arrangement of UI elements in the SaaS application is convenient and logical.
- Assess the content layout for user-friendliness and smoothness.
- Test if the number of usage steps to achieve a certain result in the interface is appropriate.
- Check if the SaaS solution complies with the conventional usability standards.

# 2.2.7 Localization testing

- As a SaaS application can be used from any location in the world, verify that the solution supports international characters.
- If the application supports multiple target languages, check that their implementation is properly optimized.



# 2.3 Challenges of SaaS testing

## DELIVERY SPEED

SaaS solutions are released and upgraded at a faster speed than traditional software. So, testing activities should be performed quicker and more frequently as well. Ensuring such speed can be a challenge for QA engineers.

## SECURITY ISSUES

Security should be a constant concern for SaaS providers, and testing plays an important role in ensuring there are no issues in this area. During SaaS security testing, taking into account various privilege levels, privacy requirements, and behavioral patterns of multiple individual users can be challenging. Verifying and ensuring security and privacy also become particularly important and difficult during migration and integration, as well as during the periods when testing and production can mix.

## PERFORMANCE TESTING

Flawless performance is one more crucial requirement for SaaS solutions. Effective testing of SaaS application performance can be more challenging than with traditional applications as proper performance should be checked for many more users from different locations. Another difficulty is identifying the most accessible areas.

Click the section to jump ahead

Introduction
1. The Software as a Service (SaaS) model
1.1. What is SaaS
1.2. SaaS specifics
1.3. SaaS advantages
1.3.1. Advantages for SaaS users
1.3.2. Advantages for SaaS providers
1.4. SaaS challenges and risks
1.4.1. Challenges and risks for SaaS users
1.4.2. Challenges and risks for SaaS providers
2. Testing of SaaS solutions
2.1. Specifics of testing SaaS software
2.2. Checklist for SaaS testing
2.2.1. Functional testing
2.2.2. Security testing
2.2.3. Performance testing
2.2.4. Compatibility and Interoperability testing
2.2.5. API testing
2.2.6. Usability testing
2.2.7. Localization testing
2.3. Challenges of SaaS testing
2.4. Best practices for SaaS testing
Conclusion
About QATestLab
Contacts

## LICENSING

With each new release of a SaaS application, QA engineers need to test all the licensing factors such as the solution's current functionality, number of users, and usage (for example, the amount of specific data or volume of transactions).

## LIVE UPGRADE TESTING

Before new releases, testers need to verify that live upgrades will not have any negative impact on the existing SaaS users. It adds more tasks for the QA engineers checking a SaaS application and can be more challenging than with traditional applications because of the interconnectivity of Cloud users and many extra factors that need to be taken into account in a SaaS solution.

## TESTING BACK-END COMPONENTS

QA engineers checking a SaaS application occasionally need to test the dependencies between the application and back-end components linked to the SaaS users' interface. It can be challenging because these back-end components often cannot be reached by testers as they don't have access to the source code.

# 2.4

# Best practices for SaaS testing

## TEST AUTOMATION

Fast and frequent software updates, which are an integral part of the SaaS model, require efficient and quick testing activities. Test automation can help in this situation. Automating repeatable and long-running, as well as data-intensive tests can optimize the testing effort. However, automated tests cannot completely substitute manual testing. So, as always, the right balance of automated and manual testing is required for the best results.

## CONTINUOUS INTEGRATION / CONTINUOUS DELIVERY (CI/CD) METHODOLOGY

When the approach of frequent and small releases typical of the CI/CD methodology is applied to SaaS testing, it allows to deal with the speedy updates of SaaS software easier and better. It also allows to find more bugs, especially since continuous functional and regression testing are an important part of the CI/CD approach to quality assurance.

## RISK-BASED TESTING

Since time is of the essence in SaaS testing, focusing the testing effort in the areas that are the most crucial for the SaaS solution is a reasonable strategy. To determine such areas, it is necessary to research which functionality customers use the most and consider the most important. Another essential factor is to determine in which areas possible issues would have the worst effect on the SaaS company. All these areas will need to be tested particularly thoroughly.

## A PROPER TEST ENVIRONMENT

When testing SaaS solutions, using a production environment is usually not an option because it will have a negative effect on the existing SaaS customers. In this situation, setting a proper staging environment is very important. It will allow to get reliable test results without influencing the existing application users during the process.

## SERVICE VIRTUALIZATION

When testing integration of a SaaS solution with various external dependent components (for example, SaaS users' databases or enterprise software), these external components often cannot be reached at the time of testing. In this situation, service virtualization can help. It simulates the dependencies between the SaaS system and the external components in question. This is particularly important for functional, performance, and API testing.

### POWERFUL HARDWARE

Using powerful hardware during quality assurance of a SaaS solution helps to verify its performance with additional resources.

### INCREASING THE WORKLOAD DURING PERFORMANCE TESTING

It is useful to add concurrent users from multiple environments during performance testing of a SaaS solution to increase the workload. This way testers can better verify the stable performance of the application under different conditions.

### THOROUGH SECURITY TESTING

Security testing is one of the main priorities when it comes to quality assurance of a SaaS application. One of the important things to note is that security also always needs to be tested during migration and integration.

# Conclusion

Compared to the traditional software delivery model, SaaS has many differences and unique characteristics. They lead to multiple advantages and benefits of SaaS, but also to a number of risks and challenges associated with it. Testing SaaS solutions is usually more difficult than testing traditional software, but there are ways to overcome these difficulties and optimize the testing process. To do so, one should follow the best practices of SaaS testing.

We hope that this paper answered all the questions the readers may have had about Software as a Service and testing SaaS solutions. Our description of SaaS specifics and pros and cons should help those who consider using SaaS applications or starting a SaaS business to make an informed decision. The information about the challenges and best practices of SaaS testing, as well as a comprehensive testing checklist will be useful for QA experts who want to optimize their testing of SaaS solutions.

If you would like to get more information about SaaS testing, have any questions, or need assistance in testing a SaaS solution, please contact us. The experts at QATestLab will be happy to help you.

# About QATestLab

QATestLab is a leading software testing provider with more than 15 years of experience. It helps companies and product owners assess the quality of their software to ensure great customer experience and profitability.

With more than 250 QA engineers skilled in manual and automated testing, QATestLab can cover all the testing needs. We test, consult, and provide QA management services. Our test lab includes 350+ real devices for assessing the work of software in real environments.

**QATestLab helps to verify the quality of SaaS solutions by:**

- Checking the work of the product's existing and added functionality
- Evaluating the performance of the solution
- Assessing the cross-browser and cross-platform compatibility of the software
- Detecting security vulnerabilities
- Checking if the solution is user-friendly
- Testing APIs
- Performing and assisting with test automation, and more

Contact us to find out more about the benefits of collaboration with QATestLab and the ways of cooperation with us.

# Contacts

E-mail: contact@qa-testlab.com

Phone: +357 22 27-04-66

Headquarters address: 34 Falirou Street, 1st Floor, Office 115
2066 Strovolos, Nicosia, Cyprus

Website: qatestlab.com

LinkedIn: QATestLab

Facebook: QATestLab

SkypeID: sales.qatestlab