



Codeless Testing Automation



**CONQUERING THE CHALLENGES
OF CONTINUOUS TESTING**

INDEX

INTRODUCTION	02
ABSTRACT	03
UNDERSTANDING CONTINUOUS TESTING	04
CONTINUOUS TESTING CHALLENGES AND THEIR SOLUTIONS	06
CONCLUSION	09

INTRODUCTION

The digital world is one of the fastest changing spaces, with innovation constantly taking giant strides. One of the key drivers of this rapid innovation is the ever-changing consumer preference. Keeping pace with these expectations is important for every business because today the digital consumer is spoiled for choices. In this highly competitive space, brands are difficult to build and equally difficult to sustain. Superior viewer experience is the name of the game, and with this comes the need for speed, accuracy and security.

Rapid innovation and high competition can be a lethal combination for the laggards in the software arena. A corollary of this double whammy is highly diminished testing time, with no room for compromise on speed, accuracy and security. But then, digital innovation also includes innovation in the software testing space. One such innovative concept is Continuous Testing which was born from the need for rapid testing, quick feedback and early rectification. This in turn helps achieve the twin objectives of getting the app or its updates to the market before competitors, and also ensuring superior viewer experience.

Continuous Testing which promotes faster go to market time, is imperative for every app because in this highly digitized world, the app is the company's link to quickly reach current and potential customers. Yet according to a Forrester Research report, only 26% of organizations truly follow the tenets of Continuous Testing. Many start off with enthusiasm which slowly wanes when issues are faced. These issues may arise from a lack of proper planning, from complexities encountered or from the overwhelming quantum of results that need to be analyzed. This whitepaper aims at presenting a holistic view of Continuous Testing and providing pointers on how to conquer the challenges, so as to reap the benefits that well sustained Continuous Testing offers.



ABSTRACT

Gone are the days when software testing was the tail-end of the software development life cycle (SDLC). Today app development and testing need to go hand in hand as the time available is highly crunched. In this scenario, Continuous Testing if implemented properly can offer many benefits including speed, accuracy and security. It also provides a good platform for promoting Devops and Agile Methodologies which are so vital in this highly digitized world where innovations take place at break-neck speed. A report by Sauce Labs states that 97% of organizations have adopted Agile and 71% are practicing or adopting Devops. Hence it's imperative for software teams to adopt Continuous Testing which greatly supports these widely used technologies.

This whitepaper has been presented in two parts. The first will be a brief understanding of the concept of Continuous Testing and the second part will cover the challenges faced in its implementation, as well as solutions for overcoming them. A brief outline of these two sections is given below:

Understanding Continuous Testing

This section will briefly touch upon the following:

- Concept of Continuous Testing
- Components of Continuous Testing
- Essentials of a Good Continuous Testing System.

Continuous Testing Challenges and their Solutions

This section will explore the following challenges and their solutions:

- Logjams due to Underestimated Time and Resources
- Technical Complexities
- Analyzing the Overwhelming Results

It is hoped that this reading will help alleviate the hesitancy faced in implementing Continuous Testing.



UNDERSTANDING CONTINUOUS TESTING

Continuous Testing is the process by which software code testing is automated at every stage of the SDLC, permitting the commitment of new code modules as they get generated. This ensures early detection and rectification of errors.

Continuous Testing is a great enabler for development and testing teams to work together throughout the development phase. Developers can test features and functionalities as they develop, since their updates go through test automation and continuous integration, to ensure that the new codes do not negatively impact existing approved functions.

If properly implemented, Continuous Testing can save a lot of time, energy and resources, which in turn translate into cost saving – that vital factor which is so important to every business.



A review of the components of Continuous Testing will offer deeper insight into what it entails and provide the context in which the challenges arise.

Continuous Testing basically consists of the following components:

01

Test Automation

Test automation is vital for Continuous Testing and comprises of using suitable testing tools to automate the software testing process, thus minimizing human intervention, with a view to increasing speed and accuracy.

02

Continuous Integration (CI)

This entails collecting the codes of the many developers involved, and placing them in a common code repository, in order to check whether these codes are correctly built and tested.

03

Continuous Delivery

This is the process by which the server gets the new code from the repository, builds it and tests it to check if it can be merged with the rest of the codes.

04

Continuous Deployment

Here the updated builds which are small, frequent, thoroughly tested changes in the software, are kept fully ready and then deployed to the production server. The functioning of the app needs to be scrutinized after every deployment.

05

Production Testing

Production Testing is the process by which software is tested in the production environment before it is released to the end user, and continues into live testing in the actual production environment.

It is apparent, that every component of continuous testing is important and needs to be accurately done, as the components form a chain that leads up to the final success of the app.

The Essentials of a Good Continuous Testing System are:

- **The Right Continuous Testing and Integration Tools**
- **The Right Automated Testing Environment**
- **End-to-End Testing**
- **A Well Co-ordinated Team.**

A more detailed reading on these essentials and on Continuous Testing in general can be accessed from our whitepaper on 'Introduction to Continuous Testing'.

This background on the concept of Continuous Testing provides the context in which the challenges are encountered. The next section will deal with these challenges and explore how they can be overcome.

CONTINUOUS TESTING CHALLENGES AND THEIR SOLUTIONS

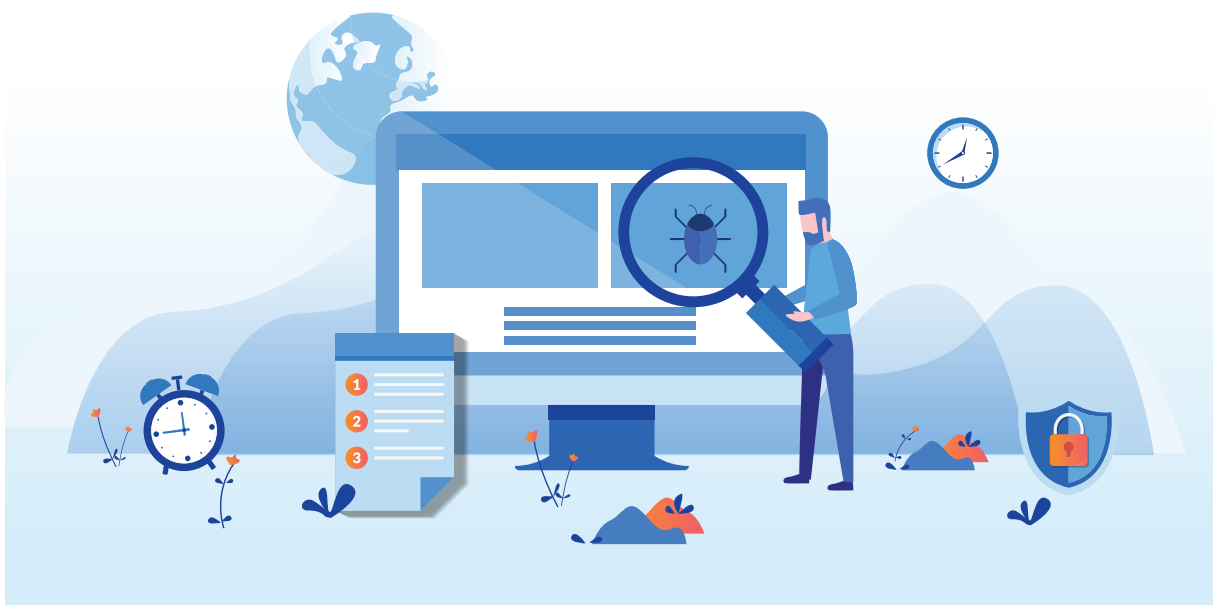
Continuous Testing is indeed imperative to meet the challenges of speed, accuracy and security of apps within the highly crunched testing time available. Yet as mentioned earlier, statistics show a very low adoption rate of just 26%! What is the cause of this hesitancy to implement this important methodology?

This section will provide insights into the challenges faced and will also explore how to counter them. Overcoming these problems is the key to reaping the benefits of Continuous Testing in terms of avoidance of errors; elimination of testing bottlenecks and costly delays; increasing testing speed; avoiding expensive rework; and ensuring faster go to market time.

Before expounding the challenges, it is important to understand the goal of Continuous Testing, which goes far beyond mere automation of software testing. While automation throws up a list of passed and failed codes and integrations, Continuous Testing, pays special attention to the qualitative aspect of these results, and checks on various aspects of the app release vis-à-vis the business risks. It aims at harnessing service virtualization; ensuring that test data management is where it should be at each stage of the SDLC; stabilizing testing for continuous integration; and promoting exploratory testing to detect major issues at an early stage of iterations.

Successful Continuous Testing mandates that the essentials that were touched on in the earlier section must be in place. There should be a harmonious blending of the right tools, processes, technologies, methodologies and most importantly, teams that support and enhance the tenets of Continuous Testing. In other words, transformation is required in technology, testing environment and in the team thinking, for Continuous Testing to succeed.

Even with the essentials firmly in place, there may still be challenges when implementing Continuous Testing. These are presented below, along-with ways to counter them.



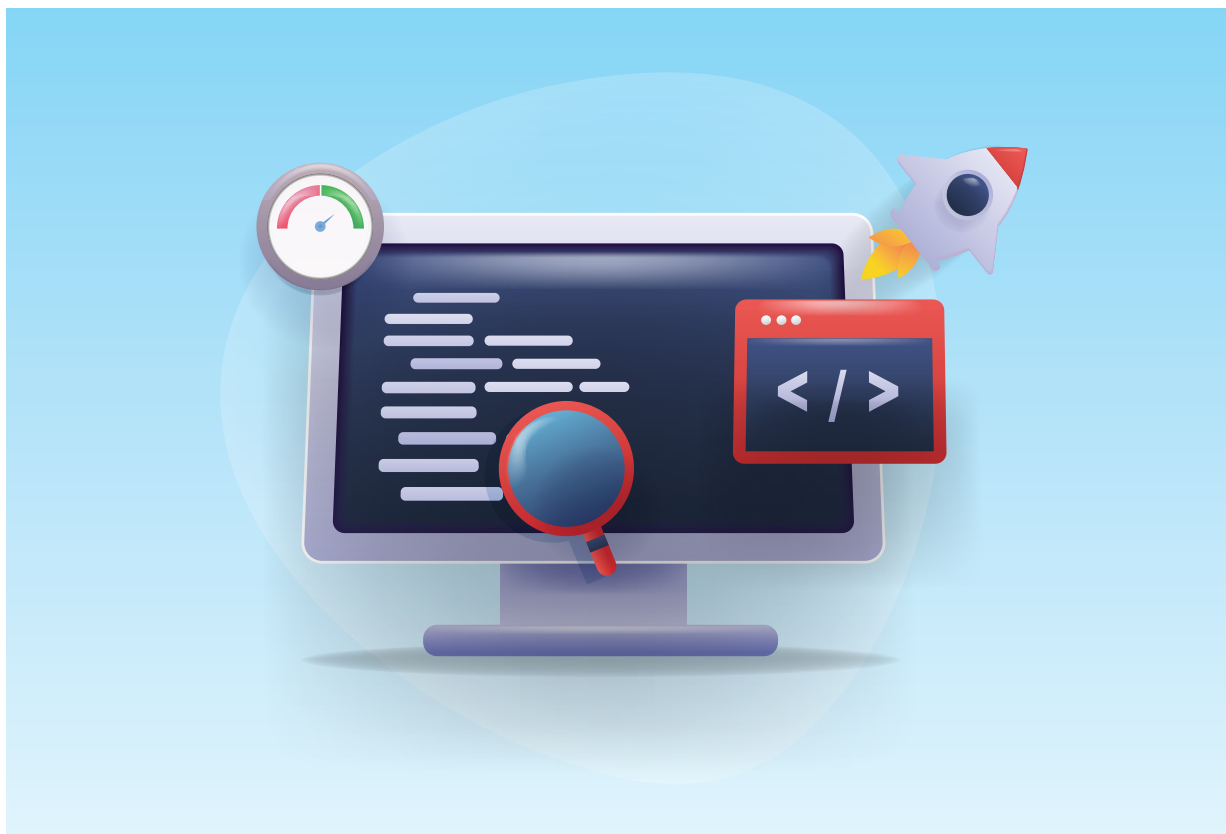
1. LOGJAMS DUE TO UNDERESTIMATED TIME AND RESOURCES

Moving over to Continuous Testing necessitates automation of processes, and testing early and continuously in the SDLC. However there is tendency to underestimate the time and resources needed and this results in logjams in the testing process.

Solution to tackle this challenge

- Keeping in mind that time is of essence, the focus should be on validating only robust test scripts and avoiding the weak ones which unproductively burden the resources. Prioritization is required to determine where efforts are needed and where they can be avoided.
- A test agenda that supports reuse and data-driven testing must be put into place. This will ensure enduring and sustainable test automation and save time and resources.
- Individual tests and the broader test framework should be kept continuously in sync with the evolving application.
- A good place to start is the execution of large and UI-heavy test suites, automating the more advanced use cases and running them in a continuous testing environment.
- Mounting test results should be evaluated and analyzed frequently to keep them from overwhelming the system.

In short: Time for test creation should be well managed, resources should be maintained, test suites executed, test results analyzed, and quick feedback ensured through an ongoing feedback mechanism.



2. TECHNICAL COMPLEXITIES

Technical complexities arise with the fact that the app has to align with various technologies existing in the organization. To optimize the critical factors of time and resources it is important to automate wisely, focusing on the most business-critical transactions, which go through multiple technologies like SAP, APIs, Mobile Interfaces, and Mainframes, to name a few. This requires a high-tech set up and detailed planning, as has been elaborated below:

Solution to overcome this challenge:

- Testing resources should be made aware of how to automate tests across all the different technologies involved and should be able to connect data and results between these various technologies.
- Care should be taken to ensure that the test data is as relevant, secure, and compliant as is required to set up a realistic test; as well as drive the test through a complex series of steps every time it is executed.
- It's also important to put in place dependable, continuous, cost-effective access to all connected systems which are required for effectual testing. This should include access to APIs and third-party applications that may be unstable, evolving, or accessible only at limited times.
- Systems should be set up to efficiently eliminate critical defects related to user experience. Superior viewer experience makes that vital difference between apps that thrive and those that fade out. This may be something that is beyond the purview of automation and may require human intervention.

In short: It's important to put in place a strong and reliable mechanism for assessing the app, focusing on the users' perception, so that serious defects are weeded out and technical complexities are moderated.

3. ANALYZING THE OVERWHELMING RESULTS

The main factor that dissuades organizations from following through on Continuous Testing is the sheer volume of results! There's a tsunami of numbers w.r.t. tests that have passed, failed or did not execute. However, these numbers do not provide qualitative data pertaining to the impact vis-à-vis the business related risks. There may be many false positives. Large failure percentages may pertain to something very insignificant, whereas small percentages may relate to a functionally major business risk. Furthermore a critical functionality may not have had the meticulous testing it needs. The tedious, analysis that is required is often a put-off, leading to testers ignoring false positives. This is more so when the test suites become very large and test frequency increases tremendously. Dealing with false positives time and again in highly crunched timeframes can be quite overwhelming and as a result the false positives are ignored, thus throwing out the baby along-with the bath water. This is extremely risky and can greatly compromise the testing quality.

Solution to this challenge:

- Manual assessment of test results is essential to ensure that critical functionalities are addressed. Evaluating the test results w.r.t. the business risks is time consuming, but essential. There will be the temptation to shelve this detailed analysis because of the volumes. However this exercise is critical for safeguarding the efficacy of the app and also for generating confidence before the app is released.



CONCLUSION

Continuous Testing is a great boon though it does come with its fair share of implementation issues. But these are not un-surmountable, and in the long run, the benefits of Continuous Testing far outweigh the comparatively few issues involved. Rapidly changing customer preferences, compressed testing time, and the need to get to the market before competitors, necessitates a move to faster technologies and methodologies. This is also the reason for the sweeping transition to Agile and Devops. Continuous Testing goes hand-in-hand with these practices and is therefore the right way forward to ensure that the three pillars of software testing viz. speed, accuracy and security are not compromised. Investing time and energy into Continuous Testing will hold the organization in good stead in the long run. But there should be a wholehearted implementation of Continuous Testing if the benefits are to accrue. The right tools, the right testing environment and well-coordinated teams that have the right temperament for Continuous Testing, will greatly contribute to the app's success in a short turnaround time.

For Mobile App Testing, BOTm is a great option since its in-built technology supports the vital methodologies of Agile, Devops and Continuous Testing. BOTm's AI and ML driven automated testing ensures quick, accurate and secure testing across spectrum. Visit www.botmtesting.com and sign up for a **Free Trial**, to experience ease in mobile app testing.

GET IN TOUCH

 **022 4050 8200**

 **sales@botmtesting.com** |  **www.botmtesting.com**

BOTm is the accelerator BOT for automated and manual testing of mobile applications -
developed for both Android and iOS devices.