

AUTOMATORS

BELIEVE IN TRANSFORMATION

**CONTINUOUS
TESTING ROBOTS**



CONTINUOUS TESTING ROBOTS

The speed of digital disruption has mounted pressure on developers and testers. Anyone in the software products space knows that traditional (classical) ways of testing and delivering software's and applications are no longer sustainable, neither are they adequate. Unlike long ago when companies used to release software updates quarterly, bi-annually, or annually; today iterations commonly last 2 weeks or less. One of the compelling innovations in recent times to counter this rising demand for delivering quality yet complex products within short delivery cycle time is software test automation.

1972

HISTORY OF SOFTWARE TEST AUTOMATION

Well, software test automation involves the use of a subsidiary software that is separate from the one being tested to manipulate the execution of tests, and to make comparisons of the actual and predicted outcomes. In simple terms, it is typically a tool to aid testing.

Test automation isn't a new concept in the computing industry. In fact, it dates back to 1972 when the first conference regarding software testing was held in Chapel Hill. It was discovered that programming was not the only discipline worth studying. Testing was also an important discipline to be studied. However, test automation wasn't fully achieved after this meeting due to poor understanding of the subject.

Later truckloads of diskettes were sold by automation proponents with an aim to put an end to manual testing. However, this too hit roadblocks as automated tests had to change every time the application being tested changed.

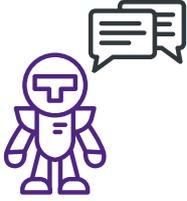
Users later discovered that despite the fact that they had automated libraries, a serious update was necessary for automated test suits to work. Test automation hit a roadblock, and all the testers were put away in boxes.

The enthusiasm of automation never faded away. Its proponents later came up with new and innovative ways to come up with testing tools. Various experiments and the explosion of the internet pushed test automation back into the limelight.

Test automation is currently the most important tool in building a grand, yet stable and strong application software more so in an agile type environment. Automation in software testing is predicted to be the most sought-after trend in application and product development cycle. Automation of risk-based testing, shift-left testing, technical debt reduction and early testing are all super-promising prospects in software development.

One of the greatest agility of software automation is achieved with continuous testing.





CONTINUOUS TESTING

Continuous testing involves conducting automated tests on a portion of the entire software delivery pipeline with an aim to acquire relevant feedback on the risk linked to the release of a software as rapid as possible. It constantly evolves and stretches test automation to address the ever-increasing complexity and speed of modern application delivery and development.

It is meant to reduce the time taken for developers to get feedback by bringing in traditional developer or tester-triggered tests as well as environment-triggered tests. Continuous testing also provides an insight on whether releasing a candidate to the delivery pipeline is too risky.

We have seen how testing has evolved from the old testing techniques – classical testing – to newer and advanced continuous testing strategies. While continuous testing may seem like an end-all solution, one may want to know, *how different is continuous testing from classical testing?*

Well, let's take quick a quick look at the two.

CONTINUOUS TESTING VS. CLASSIC TESTING

Ideally, these two testing approaches have their own unique competencies.

i. Exposure of candidates

The moment a business decides to expose a candidate to the end user, they risk software fails on a large-scale level.

Continuous tests are built with risk in mind which gives the businesses insight on how to assess the situation.

Classic tests are valuable, but they barely stop high-risk candidates from getting into the market.

ii. The breadth of the tests.

Continuous tests are broad enough to detect a change in the application that would negatively impact the functionality.

Classic tests are not as thorough, and businesses risk their public image when a candidate doesn't meet the consumer's expectations.



WHAT ARE THE CHALLENGES OF CONTINUOUS TESTING?

⚠ Automation rate

The current automation rates (10-30%) are way too low to enable fast feedback needed for continuous testing. For efficient continuous testing, automation rates need to be over 80%.

⚠ Test Data Management

Many test cases fail because test data is not readily available.

⚠ Regression maintenance

When applications get new features usually parts of the automation breaks, due to the bad buildup of the test cases. It takes too much unnecessary time to maintain the automation.

⚠ Progressive automation

New features are often tested manually; nobody ever takes care about automating them. You soon end up with too much functionality which is not automated.



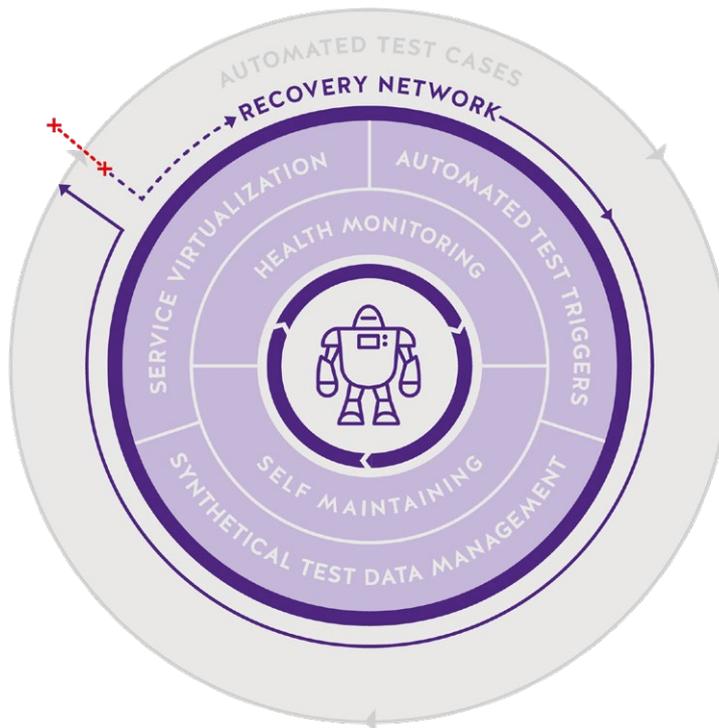
- ⚠️ **Continuous execution**
There is the need to execute test cases continuously. Due to missing data or unstable system environments continuous execution is often not possible.
- ⚠️ **Business risk coverage**
Only a few people know what the most important functions of an application or system are. Their knowledge is often not accessed, or they don't have time to communicate with the testers to make sure that the right areas are tested.
- ⚠️ **Waste reduction**
Too many irrelevant test cases are created wasting the time of the employees and creating unnecessary time for maintenance. Managers are not aligned, and double effort is created.
- ⚠️ **Infrastructure**
Infrastructure is not set up properly to ensure a staged delivery pipeline. Test environments are too unstable, and people are blocked because of that.



At Automators, we understand that fast paces of modern software development have created tremendous challenges for developers and the test team; hence the need for different strategies in automation. By following the imperative of “test smarter, and not harder,” Automators have borne the continuous testing robot for quality assurance.

WHAT IS A CONTINUOUS TESTING ROBOT?

A continuous testing robot or a CTR is an intelligent software solution which runs complex automated test cases. It has a built-in recovery network. It provides synthetical test data management that helps you connect the right data to your test cases and creates new data when necessary. No test case should fail due to test data issues. Due to its modular structure, you can maintain and add new test cases very fast.



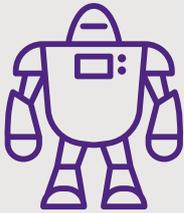
When using automated test triggers, it can be run continuously. A risk-based requirement structure tells you how much of your business risk is covered and what to focus on. When appropriately enhanced it has a high automation rate, and parts of the infrastructure can be simulated when needed. A health monitor will keep you up to date on the quality of your system/application.

Here's a more expansive look at the various areas where CTR is being advanced in testing:

APPLICABILITY OF CONTINUOUS TESTING ROBOTS

1. **Live Health Monitoring** – displaying results of the test cases.
2. **Automated Test Triggers** – automatically running test cases at any predefined time.
3. **Service Virtualization** – simulating parts of the environment.
4. **Automated test cases**
5. **Recovery network** – resetting the environment if test cases fail unexpectedly.
6. **Synthetically test data management** – creates and finds data within the testing system. can be connected to the recovery network.
7. **Self-maintaining** – test cases are built with dynamic IDs. Changes in the application don't always break the Test cases.

THE WRAP-UP



The future of automation testing and CTR looks bright. With the development of new automation strategies/tools and incorporation of AI in testing, the ground is set for the new competitiveness and industry disruptions.

The orientation of continuous testing and CTR and continuous products delivery pipeline within an enterprise will ultimately determine the long-term competitiveness, success, and standing of the final product/application in the market.





CONTACT DETAILS

AUTOMATORS GmbH

Saturn Tower
Leonard-Bernstein-Straße 10
1220 Wien

+43 1 393 00 50
office@automators.com

automators.com

