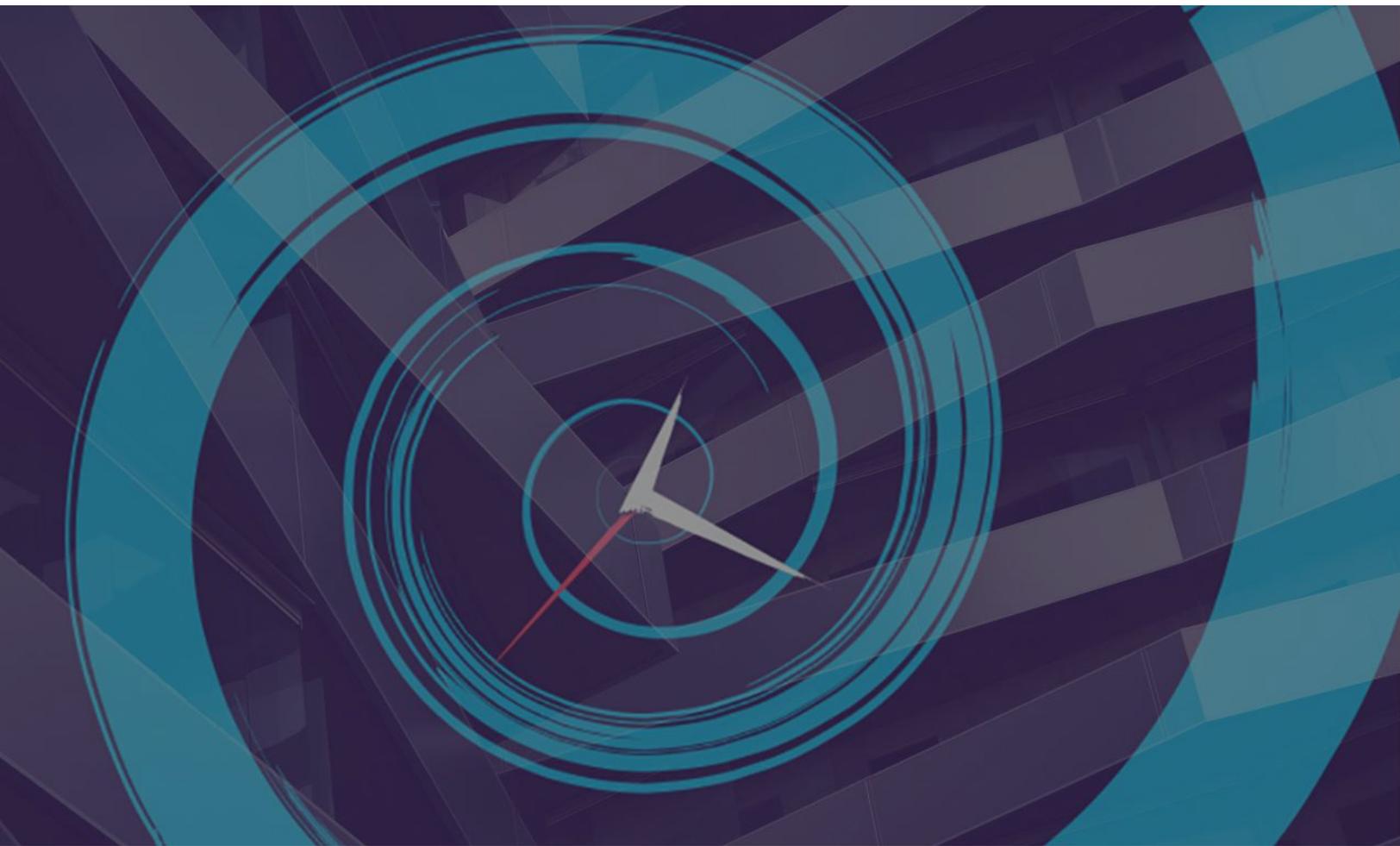


Accelerate CI/CD Pipeline with Continuous Testing

Authors: Anuradha Muralidharan & Jagdish Anandan



Introduction

Most enterprises have already implemented Agile methodologies to speed up their software deliveries. As a testament to its usefulness, enterprises embrace Agile not only for development but for other functions such as IT and operations to streamline processes, improve feedback, and accelerate innovation.

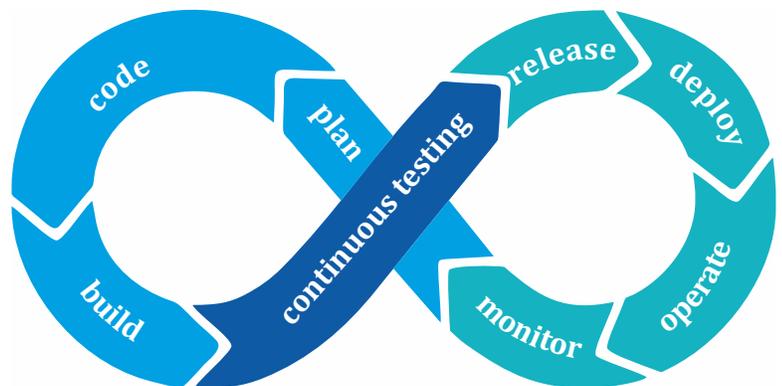
Taking Agile to the next level, organizations see DevOps as Agile version 2.0. DevOps has now led to Continuous Integration and Continuous Delivery (CI/CD). While Agile helped to streamline and automate the entire software delivery lifecycle, CI checks the code often, and the tested chunks are integrated, sometimes several times in a single day, to create a stream of smaller and frequent releases through CD. Kurt Bittner, a principal analyst at Forrester Research, puts it succinctly: "If agile was the opening act, continuous delivery is the headliner."

To ensure CI/CD works well, the not-so-talked-about link is Continuous Testing (CT) which has emerged as the catalyst for enabling quality and speed. With its Shift-Left approach, CT has been a central force to enable CI/CD practices.

What is Continuous Testing?

- Continuous testing (CT) is like an armor in the CI/CD pipeline process. It buffers the developers by squashing bugs. It ensures that end-user experience remains uncompromised no matter how many releases happen.
- CT can also be called a safety net, as it protects the end user experience in spite of accelerated processes.
- Since it is a continuous process, it cannot be tagged at the end, but needs to be seamlessly interwoven into the software delivery pipeline.
- It establishes a stable test environment with valid test data at all stages of development.
- Tests should be wide to adapt to an application's change, which affects functionality.

CT requires tools, teams, individuals, and services. Even if CI and CD ensure faster time to market in the absence of a thorough end-user experience testing, the solution may become a liability rather than an advantage. CT, as the nucleus, is essential to ensure that all things endless, continue flawlessly. **To ensure that continuous testing does not impede the pace of CI/CD, test automation is required.**



Digital disruptors and customer-obsessed organizations are improving customer experience (CX) by shortening their software delivery cycles, delivering features in smaller increments, and scaling their existing Agile processes in concert with DevOps. Traditional testing services don't cut it for these organizations: 20 of 25 reference customers told us that they are adopting continuous testing (CT) services to support their Agile and DevOps initiatives within a digital transformation journey. Of those CT services, clients say automation is the most impactful and differentiating for delivering better software faster.

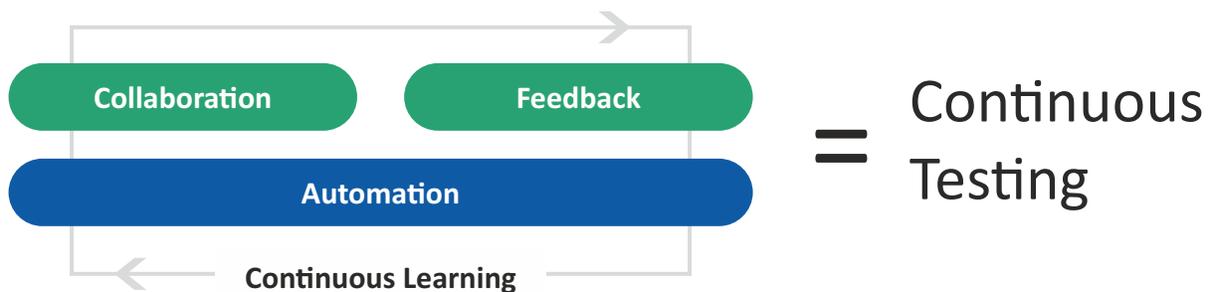
- The Forrester Wave™: Global Continuous Testing Service Providers, Q1 2019

Test Automation – CT’s Secret Success Factor

Automated testing is key to the successful integration of quality assurance into DevOps workflows. It is the only way to ensure that quality assurance is as continuous, agile, and reliable as the rest of the operation. CT involves automating the tests and running these automated tests early and often, along with service virtualization to set up all the required infrastructure. Automated testing must be integrated at the initial stages of the development cycle to ensure CT.

In this sense, automated testing serves as the glue that binds together all of the other processes that comprise the continuous delivery pipeline. Without automated testing, DevOps doesn't work. The World Quality Report emphasizes the need to increase test automation. To deliver software at speed requires smart automation and intelligence. This will help to validate continuously changing scenarios.

A test automation framework is the building block to a continuous testing approach. When you build off automation with collaboration and instant feedback, continuous learning begins to surface, and teams can start to capture the real benefits of continuous testing.



Why Automated Testing is Essential for CI/CD

- Automated testing allows us to execute tests in parallel, across multiple servers/containers, speeding up the testing process.
- There is more consistency in automated testing. Automation ensures that software behaves the way it is expected to and reduces human errors or bias.
- In a CI/CD pipeline, tools and frameworks have to be adjusted quickly to meet changing demands. Manual testing makes it tough to update and be agile. For example, if you switch to a new CI server platform, the entire manual testing process has to be reinitiated. When you have automated tests, however, most of the configuration is done automatically. As a result, you can migrate to new environments quickly.
- Maximizing workforce is crucial in a development project. Repetitive testing steps, when automated, free up the engineers to focus on other high-value tasks.
- When changes are small in a CI/CD pipeline, they still need all the effort that goes into testing. Automated testing works well to validate minor changes continuously.

Top 5 Automation Testing Tools in Continuous Testing

1 **Testsigma** is a SaaS software using a Shift-Left approach. It uses natural language statement to produce steps and is easy to use for manual testers. Writing tests is easy and helps save time.

Testsigma requires a mature test environment and infrastructure that supports continuous testing. It has several features, such as Test Data Management, Object Repository, and integration with CI/CD tools.

2 **Selenium** is an automated testing tool used extensively for web application testing. It supports multiple operating systems and browsers along with various programming languages such as Java, Groovy, Python, C#, PHP, Ruby, and Perl.

For Continuous Testing, Selenium may not be sufficient, and other tools such as Robot Framework and similar options need to be considered. However, integration can be a big challenge.

There are many other test automation frameworks (open source and commercial) similar to Selenium available on the market today.

3 **IBM Rational Functional Tester** is beneficial for storyboard testing and test scripting. It offers API, functional UI, and performance testing. It enables Shift-Left testing and helps developers by identifying bugs early in the development cycle.

4 **Tricentis Tosca** does a great job in test coverage in automation testing. It provides feedback and helps to fix bugs early. Using a model-based approach, it helps to reduce maintenance costs. A Model is an abstraction of a functionality. When changes are made to the model, Test Cases are automatically synchronized to fit the new model.

5 **Unified Functional Testing(UFT)** has advanced image-based object recognition features and reusable test components. These help to reduce redundancy and help to convert manual tests to automated tests. Test scripts in UFT are crafted using Visual Basic Scripts which requires a considerable amount of scripting knowledge.

Characteristics of good Automation Framework

- Codes keep changing, and that is a fact of life. Development processes too change, i.e., web page, to the hybrid app and so forth. Since the future of technology, to a large extent, remains unpredictable, the test frameworks should support various languages, browsers, and techniques. Maintaining agility in a continuous delivery pipeline requires an agile testing environment
- The testing platform should be able to scale to support as many tests as needed.
- It is essential that the platform supports compatibility tests on simulated devices. This helps to cut down the cycle closer to the last mile.
- The environment should maximize automation, i.e., trigger tests, analyze results, and share test information across the organization, in a fully automated manner.
- Cloud-based testing also allows you to avoid false failures generated by on-premise test grids, as well as the inefficiencies that can arise from having to find and fix bugs in on-premise test infrastructure.
- The testing platform must include security features such as encryption of test data and access control policies.

How to Identify Tests for Automation?

It is impossible to automate 100% of tests - manual testing might be more appropriate, in some instances. To have a tester, manually test something that can be automated is a waste of tester's potential. A broad rule of thumb will be to automate all those tests that are executed frequently. These tests, when running manually, can be prone to errors, based on the fact that the human brain can become weary of seeing the tests multiple times in a day. For example, cross-browser compatibility testing is one function which can be easily automated. Ensure all your automated tests are reliable and do not generate false positives - due to uncontrolled starting state or environment variations, etc.

The second rule would be to automate all those tests that are dependent on a person. Dependency on developers and testers is risky if the concerned personnel are not available during a crucial phase. A testers absence will affect the entire CT process.

Keeping in mind the more significant, long term benefits of automation will build the criteria around which tests need to be automated.

It is better to start with a small set of reliable, automated tests and add to them over time, instead of starting with many unreliable automated tests.

Automate Non-Functional Tests

When a new or modernized application is deployed, the application has to be completely tested to ensure performance, security, availability, and so forth. Non-functional testing helps to evaluate the application's performance under load when it is still not live. It thus covers all the tests that are not included in functional testing.

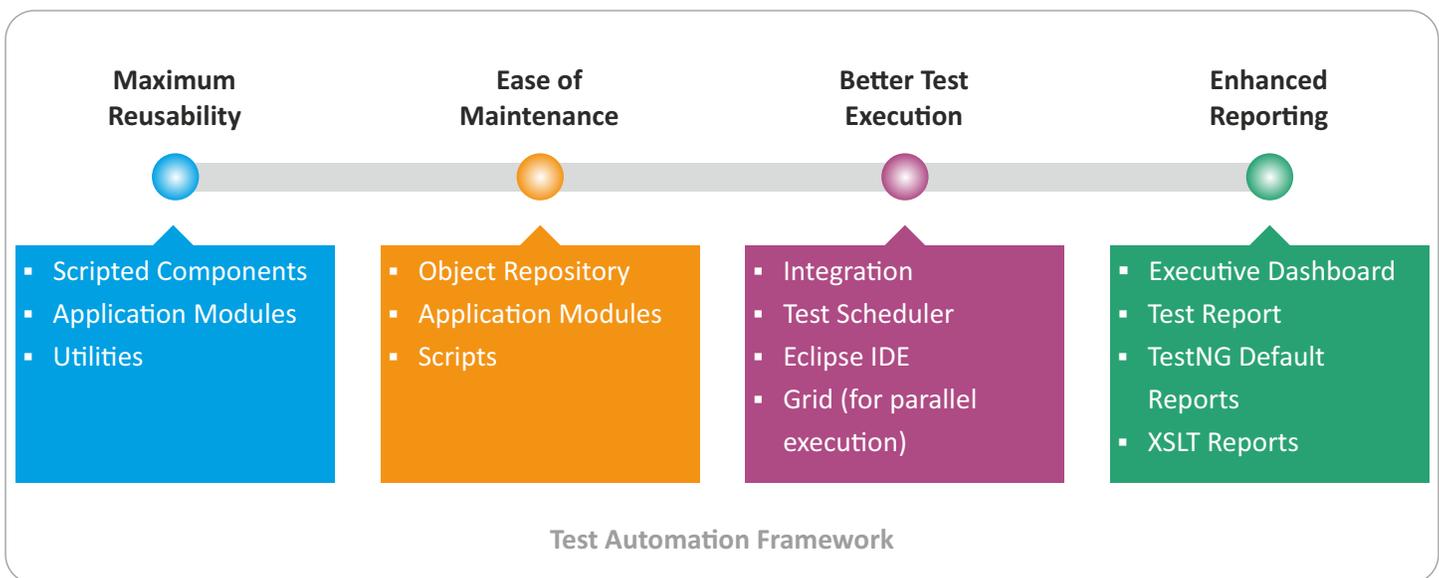
Performance testing ensures that the application will perform well and help the business to meet its goals. By looking at loads, it helps to measure response times, memory utilization, and similar system metrics. Failure to identify performance issues can lead to reduced response time and may even lead to the application failing at crucial, peak load periods. Automating load and performance testing requires planning the performance testing process, i.e., what needs to be tested, how often, etc. When goals are set with metric thresholds, test scripts are created, and finally, these are integrated with the CI tools. Similarly, non-functional testing requirements such as resilience testing, security testing, operational acceptance testing, disaster recovery testing, and scalability testing can be automated. Automating non-functional activities requires a high level of expertise to draw parallels from simulated scenarios, providing the metrics for measuring these and finally for writing the scripts which can perform these tests automatically.

Trigent's Test Automation

Reinforced by active partners such as Borland (Microfocus), is aligned with today's business environment with the ability to provide cost benefits, performance, and agility.

As niche test automation experts, we have significant experience in open source and commercial testing tools. Our extensive library of modular, reusable, and resilient frameworks simplifies scenario-based automation. We provide on-demand testing and next-gen scheduling.

Features	Benefits
Accelerated script development	Script/test cases development effort reduced up to 60-80% in comparison to traditional test automation approaches.
Modular and reusable framework components	Reduced dependency on tool-specific resources. Ability to kick-start automation quickly.
Easy test script maintenance	Ease of test execution. Easy to make changes and maintain scripts in the long run. Improved error and exception handling.
On-demand Testing	Sanity, Smoke, Integration, Regression, etc.
Hybrid Model	Modular test framework built using JUnit or TestNG.
Scheduling and customizable reporting	Send test results to ALM.



Summary

Speed or quality, CT, CI, and CD have emerged as catalysts for enabling quality at rate. All the three are dependent on superior testing to ensure frequent releases which are bug-free. Test Automation helps Continuous Testing to be agile, empowering, and business aligned.

About Trigent

Trigent is a technology solutions company that provides comprehensive solutions for business problems via outsourced software product and applications design, development and quality assurance. Trigent serves customers like Independent Software Vendors (ISVs), enterprises and SMBs in the High Tech, Healthcare, Education, Ecommerce and Manufacturing areas. Trigent's solutions help clients overcome budget, schedule and resource constraints.

To learn more about Trigent visit www.trigent.com

Our Headquarters

Trigent Software Inc.
2 Willow Street, Suite #201
Southborough, MA 01745
+1 (877) 387-4436
sales@trigent.com

Other Location

Trigent Software Ltd.
Khanija Bhavan, 1st Floor
#49, West Wing, Race Course Road
Bangalore - 560001, India
+91 (80) 2226-3000

Microsoft Partner

Gold Application Development
Gold Collaboration and Content

