

# Automated tests as part of SD and ST life cycles

**Kadola Mykhailo\*, Alexander Mrochko**

*ISMA University of Applied Science, Latvia*

*\*Corresponding author's e-mail: kadola.mykhailo@gmail.com*

---

## Abstract

Testing is a key activity in software development due to the common systems complexity, environmental variations or product specifics (e.g. based on some risks). This all causes the quality of a product and reflects the high presence of defects. Using manual or automation test activities these defects should be found and prevented from appearing again. Usually, to improve the testing process QA team applies automation activities, as example in CI/CD pipelines. But since all projects are unique and can include some risks or known problems, there is no best or the only practice or standard formulated that can be used in any project. Currently there are many frameworks in the market, or your automation testing team can even create your own and unique framework. Some of the ready to go frameworks are useful and actually can save companies human resources and money. However, using an unsuitable framework or creating an incorrect framework can lead to waste of time and resources. This work is an attempt to show the role of automation in both Software Development and Software Testing processes.

*Keywords:* manual testing, automation testing, test automation frameworks, CI/CD pipelines

---

## Introduction

Nowadays, software testing is an important aspect of the software development process. However, as the IT market has grown, the efficient and flawless operations of a product have become increasingly vital for mass-market software.

Testing is a sub-process of the entire Software Development Life Cycle, and it is a critical aspect of product development. Processes for testing should begin as soon as possible. This early start will help to improve future product quality and cost. Testing in a timely manner assists a business to save money by letting defects in the program be discovered before they enter the final product or at later stages.

The test automation technique can be used in practically any project to find flaws in the system faster and at an earlier stage of development. Test automation, like the name suggests, is the process of employing test scripts to verify the program being tested. It's usually a third-party tool, script, or framework that simulates real-world interaction with an application. To a greater or lesser extent. For example, unit tests are almost not similar to human interactions but take a short time to run, so the end-to-end tests are vice versa.

Automation tests, like development, can be done wrong, resulting in a lot of wasted time and money. As a result, the early stages of the process are crucial. Figure 1 depicts the steps.



Figure 1 Automated Testing Process

The issues are: how automation processes contribute in product development and overall testing processes including coverage of some activities with automated tests, and how they contribute to overall product quality, the pros, cons and misconceptions of automation testing and how they can be used in Continuous integration, Continuous delivery and Continuous deployment.

## Overview

This work discusses the advantages, disadvantages and conclusions on the following issues:

- Automation process as a part of development and testing processes.
- Pros, cons and misconceptions of automation testing.
- Automation in CI/CD context.
- Success Factors in Test Automation.
- Automation frameworks and Tools.
- Automation design patterns.

## Decision

Describes the role of automation in the development and testing processes while explaining some ways how it can

be applied to improve efficiency of testing process, and how to build more stable, rigour and robustness with applying design patterns.

Different testing frameworks and types of them are shown, with analysing some popular ready to use solutions. Based on personal experience shown the role of automated tests in Continuous Integration, Continuous Delivery and Continuous Deployment and the real benefits from them.

## References

[1] ISTQB Certified Tester Advanced Level Test Automation Engineer. -<https://www.istqb.org/downloads/send/48-advanced-level-test-automation-engineer-documents/201-advanced-test-automation-engineer-syllabus-ga-2016.html>

## Conclusion

Success of automation depends on applying design patterns and principles of “robust” tests and understanding of basic success factors.

Test Automation is the best way to increase the effectiveness, efficiency and coverage of software testing. Decisions to apply automation or not should be context related, based on the needs of a concrete project.

When automation testing begins, the timing varies depending on the project requirements. Depending on the project, automation in the software testing process can be attributed as both a supplementary and a key procedure, but it cannot replace all manual testing effort.

[2] Automation Testing 101: What, Why and How. - <https://www.katalon.com/resources-center/blog/what-is-automation-testing/>

[3] Test Automation Best Practices. - <https://smartbear.com/learn/automated-testing/best-practices-for-automation/>