

Analysis of Test Automation Tools: From Selection to Implementation

Arkhipov Ivan

Quality Assurance Automation Engineer, Moscow, Russia

Abstract: Automated testing is an integral part of contemporary software development, and its significance continues to grow in various domains where dependable operation is essential. This enhances software resilience against failures and malfunctions. The objective of this article is succinctly presented as providing guidance for professionals engaged in software development within the realm of testing automation, examining the stages of testing and quality assurance. The article commences with a general explanation of the process of defining the goals and tasks of desired testing automation, followed by a discussion of the key implementation stages necessary for achieving these objectives. Furthermore, the article meticulously scrutinizes a straightforward web application example to facilitate a better comprehension of these stages, commencing with goal definition, tool setup, script composition, and monitoring of the automated testing process. Additionally, the article underscores the importance of integration with the CI/CD process, which accelerates development and enhances software quality. The methodology succinctly focuses on the examination of testing automation stages, such as environment configuration, script composition in alignment with the environment, debugging for error detection, and report preservation for subsequent analysis. The primary findings elucidate these stages, which should be comprehended and utilized by any researcher. In conclusion, this article assists readers in comprehending the significance and efficacy of testing automation, selecting suitable tools and methodologies, successfully implementing them into their own projects, and ultimately achieving outstanding results in the field of programming. The article also includes discussions and recommendations for summarizing the key aspects of testing automation.

Keywords: Test Automation, Automation Tools, Programming, Software Testing, Testing Efficiency, Software Quality, CI/CD Integration.

Introduction

In the modern world of software development, where the speed of technology development is constantly increasing, test automation has become an integral part of the process of ensuring the quality of software products. Not only the speed, but also the complexity of modern applications and systems requires highly effective testing methods that can quickly identify defects and ensure the reliability of the program's operability [1]. In this context, this article becomes relevant and important, since its purpose is to provide an extensive guide for specialists involved in software development, testing and quality assurance. At the moment, efforts are focused on understanding the role and significance of test automation in modern development, as well as on describing the key criteria for selecting tools, analyzing the stages of their implementation, and highlighting best practices for the successful implementation of automated tests within programming projects [2]. This article offers readers a deep look into the world of test automation in order to help them master this important aspect of modern software development and achieve outstanding results in their work.

Purpose of the article

The main purpose of this article is to provide comprehensive guidance for professionals involved in software development, testing and quality assurance. In particular, the article is aimed at educating readers on the importance of test automation in modern software development and its potential advantages in terms of efficiency, quality and cost-effectiveness. It also aims to help readers make informed decisions when choosing test automation tools by identifying key criteria, factors and aspects that should be taken into account when choosing. In addition, the article provides an understanding of the practical aspects of the implementation of selected automation tools, including planning, configuration, scenario development, integration with CI/CD processes and monitoring, as well as highlights best practices and strategies for the successful implementation of test automation in programming projects.

The Relevance of the Article

This article is very relevant in the context of programming and software development for several reasons:

- The need of the industry: As the software development industry continues to develop rapidly, there is a growing demand for effective and efficient testing methodologies, including test automation [3]. This

article responds to this demand by providing valuable research in the selection of tools and their implementation.

- **Resource optimization:** In an era of limited resources, organizations are looking for ways to optimize testing processes. Test automation is a proven method of achieving such optimization, which makes this article relevant for specialists and organizations seeking to optimize their development processes.
- **Quality assurance:** Ensuring the quality of software products is a priority task. Automation of testing helps to improve quality by reducing the likelihood of human errors and allows for more extensive testing. Understanding the methods of implementing these practices is important to maintain high quality software.
- **Skill Development:** For programming professionals, acquiring expertise in test automation tools and practices is an important skill [4]. This article serves as a resource for developing skills for both beginners and experienced professionals.

Literature Review

In [5], the authors present the practical significance of the developed approach to automating testing of web applications. This approach has been successfully implemented in the corporate environment of Svyazkom LLC. The implementation of this approach has significantly accelerated the testing process, reduced the time required for feedback after making changes to the existing code, and also reduced the number of detected defects in the application. These improvements have had an impact on reducing the cost of developing and testing a software product, as well as improving its quality. In the context of the study of this issue, attention should also be paid to other works in the field of test automation that demonstrate similar or complementary results in improving software development and quality assurance processes.

In [6], the authors consider several important aspects of software product testing automation. Test automation is the process of using specialized software to organize and perform all stages of testing, starting from its launch and ending with analysis and output of results. Currently, automated testing has found wide application in various fields:

- **Unit testing (unit test):** This type of testing is carried out at the source code level of the program at the early stages of development and uses additional libraries such as CppUnit, Boost Test, Google C++ Testing Framework for C++, JUnit, TestNG, JavaTESK for Java, SimpleTest, PHPUnit for PHP, and is integrated into some modern programming languages.
- **Functional testing of the program's business logic (functional tests Layer):** This type of testing is aimed at verifying the functionality of the business logic of the application.
- **Testing through the user interface (GUI):** GUI automation is the most common form of automation, allowing you to simulate user actions. It is used by the "black box" methodology and does not require access to the source code of the software product.

The study also presents the main testing automation tools at the user GUI level, including playback tools, scripting and data-driven testing. Playback tools are utilities for recording and reproducing user actions and increase productivity, but require rewriting tests with changes in the application. Scripting includes test scripts written in specialized languages, and also requires changes when updating the application. Data-driven testing is based on pre-created data and allows you to run tests based on this data.

Methods

At this stage of the article, the sequential stages of the implementation of the selected test automation tools in the software product development process are considered. This process includes the following key steps [7][8][9][10]:



Figure 1: Process steps

1. **Planning:** An important stage in the implementation of test automation is the definition of a strategy and action plan. Here the goals, objectives, and scope of automation are defined, as well as success criteria are formulated.
2. **Setup:** Configure the selected tools and environment for automation. This includes installing the necessary software, configuring parameters, and creating an environment for conducting tests.

3. **Writing test scripts:** Developing test scripts that will be automated. This is an important stage where the steps, conditions and expected results for each test are determined.
4. **Integration with CI/CD systems:** Integration of automation tools with continuous integration and Continuous Delivery (CI/CD) systems. This allows you to automatically run tests with each code change and ensure continuous testing.
5. **Monitoring:** Monitoring and analysis of the results of automated tests. Monitoring allows you to identify and eliminate problems in the early stages of development and ensure the reliability of the test coverage.

These stages together form the basis for the successful implementation of test automation in the development process, contributing to the improvement of the quality and efficiency of the created software products. An example of the implementation of test automation tools in the development process of a web application.

1. **Planning:** - Defining goals: Our web application development team decides to automate functional testing to ensure the reliability and high quality of the application. - Tasks and scope: We determine which parts of the application will be automated, for example, user registration, login, adding products to the cart and placing an order. - Selection of tools: We explore the available tools for testing automation and choose the most suitable for our tasks. - Action plan: We are making a plan to create and launch automated tests.
2. **Setting up:** - Installing tools: Install the selected tools, such as Selenium WebDriver and TestNG, on the server for testing. - Configuration: Configure the parameters of the tools and specify the URL of the application for testing. - Creating a test environment: Creating an isolated test environment where automated tests will be performed.
3. **Writing test scripts:** - Test Development: Writing code for automated tests that includes steps, checks, and expected results for each scenario. - Test data: We create test data necessary for performing tests, for example, fake user accounts.
4. **Integration with CI/CD systems:** - CI/CD Setup: We integrate our automated tests with a continuous integration system (for example, Jenkins). - Automatic startup: Configure CI/CD so that the tests are automatically run every time the code is updated.
5. **Monitoring:** - Analysis of results: After each run of automated tests, we analyze the results, identify and register errors. - Optimization: We are constantly improving test scenarios and tested functions to improve the coverage and effectiveness of testing.

This example illustrates how the stages of planning, configuration, writing test scripts, integration with CI/CD and monitoring jointly contribute to the implementation of test automation in the development process and provide a reliable and high-quality web application.

An example of the practical implementation of automated tests for a web application using the Selenium WebDriver tool and the Java programming language.

Purpose: To check the functionality of logging into the web application and adding products to the cart in the online store. Practical Implementation Steps:

- Environment Preparation: 1) Install the Java Development Kit (JDK) and configure the development environment, for example, Eclipse. 2) Download and install Selenium WebDriver.
- Writing test scripts: 1) Creating a Java class for testing. 2) Import the necessary Selenium libraries. 3) Writing test methods for logging into the application and adding products to the cart. 4) The test methods use WebDriver methods to interact with web page elements, for example, findElement, SendKeys, click.

```
// Get the WebElement corresponding to the Email Address
WebElement email = driver.findElement(By.id("email"));

// Retrieve the WebElement corresponding to the Password Field
WebElement password = driver.findElement(By.name("passwd"));
```

Figure 2: example of findElement

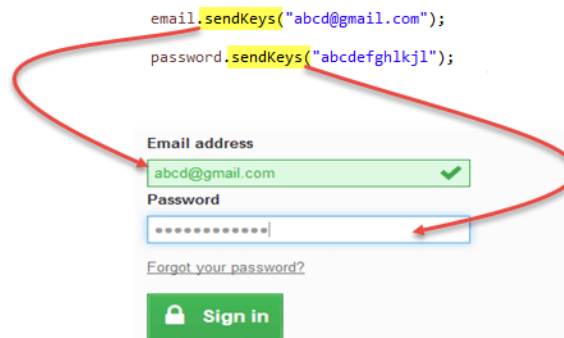


Figure 3: example of sendKeys

- **Debugging and error correction:** 1) Running test scripts to check their operability. 2) Debugging and error correction in case of unsuccessful attempts.
- **Analysis of results:** 1) Evaluation of test results, identification of errors or inconsistencies. 2) Registration of results in logs.
- **Creating Reports:** 1) Generation of test execution reports, including successful and unsuccessful tests. 2) Reports can be created using tools such as TestNG or Allure.
- **Integration with CI/CD:** 1) Setting up integration of automated tests with a continuous integration system such as Jenkins. 2) Automatically run tests when updating the code in the repository.

This is a simple example of the practical implementation of automated tests. In practice, a set of test scenarios covering various application functions is created, and more detailed configuration and optimization of automated tests are carried out to ensure full test coverage and reliable testing of the application.

Discussion and Recommendations

As a result of the practical implementation of automated tests, it becomes obvious that effective testing automation can significantly improve the software development process. However, it is important to consider several key points. Firstly, the choice of tools should be reasonable and take into account the specifics of the project. Secondly, the creation and maintenance of automated tests requires time and resources, so it is necessary to pay attention to their stability and updating with changes in the application. Finally, the integration of tests into the CI/CD process allows for continuous testing and rapid feedback, which significantly improves the quality and speed of development. It is also recommended to regularly update test scenarios, monitor and analyze the results to respond to potential problems. In general, test automation is a powerful tool for improving the quality of software, and its successful implementation requires competent planning, reliable configuration and continuous improvement.

Conclusion

In conclusion, the role and importance of automated testing were thoroughly investigated, criteria for the selection of tools and methods for successful implementation were provided, and the main stages of automation were considered in detail. Summing up, we can say that this article has achieved its goals by explaining well the stages of test automation in order to be useful and effective for other researchers in their projects.

References

- [1] Kulagin V, Sukharevsky A, Meffert Y. Digital@ scale. Desktop book on digitalization of business. Moscow: Alpina Publisher. 2019.
- [2] Krasilnikova V. Theory and technologies of computer training and testing. Litres; 2022 May 15.
- [3] Zharikova YA.V. Trends in the development of the pharmaceutical industry in the context of the coronavirus pandemic. Bulletin of the student scientific society of the state educational institution of higher education " DONETSK NATIONAL UNIVERSITY" Founders: Donetsk National University. 2022;3(14):73-8.
- [4] Medvedev O.S, Davydovsky AG, Novogran VI, Korokin LR, Yashin CD. Educational information environment for optimizing the training of specialists in the field of automated testing of graphical user interface.

- [5] Torokhov N.V. Automation of testing of the SMS center's personal account (Doctoral dissertation, Siberian Federal University).
- [6] Gordienko E.P. Evaluation of test automation and the use of automatically generated tests. In *TRANSPORT: SCIENCE, EDUCATION, PRODUCTION 2020* (pp. 25-30).
- [7] Galimova N. Decision support system when choosing a method for testing software systems.
- [8] Habibullina M.M. Web application testing: Master's thesis (2023).
- [9] Nikiforov A.V. How the implementation of continuous integration and testing helps to ensure compliance with requirements when developing a software product. *International journal of professional science*. 2023(4):88-96.
- [10] Mironova S.P. 2.10. Automated assessment of the functional capabilities of the human body: integrated monitoring technologies and information programs to ensure them. *The Development of modern higher education in Russia and abroad 2020* (pp. 279-291).