

AUTOMATION TESTING TOOLS: A COMPARATIVE VIEW

Dilara Ateşoğulları¹, Alok Mishra²

^{1,2} Atılım University; ² Molde University College (Specialized Univ. in Logistics)
e-mails: ¹ dilaraatesogullari@gmail.com, ² alok.mishra@himolde.no,
² alok.mishra@atilim.edu.tr
Turkey, Norway

Abstract: Effective software testing leads to assurance towards high quality in software development. Automation testing tool facilitates in faster testing process in testing stage thus completion and implementation of software on time. One of the most significant issues for automation is to select the automation-testing tool and the appropriate framework. The objective of this paper is to assess and compare twenty-one available automation-testing tools on twenty attributes in comprehensive manner. This study will assist software testing professionals and researchers towards further insight in this area.

Key words: Test Automation, Automation Tools, Software Testing, Automation Test Frameworks, Test Automation Types.

1. INTRODUCTION

Software testing is a process of running a program with intent of finding errors [1]. Software testing facilitates the quality of the software under development [2]. To increase the quality testing is the effective way. Manual testing and automation testing are the two methods of testing. Manual testing is also called as static testing. It is carried out by the tester. Automation testing is also called as dynamic testing [3]. As a result of this acceleration, it was necessary to accelerate the tests performed at different points and to run them in a more algorithmic manner. In the face of this necessity, software testing automations as well as manual tests have made a rapid entry into the market. Manually testing in most of the cases is time consuming, costly and chaotic. Some of the main reasons for the development of automation are listed as follows.

- Higher Testing Efficiency
- Greater Accuracy and Reliability
- Reusability and Repeatability of Test Scripts
- Improved Test Coverage

- Simulation of User Environment
- Higher ROI: Saves Time and Costs
- Volume and Simultaneity
- Early Detection of Bugs [4].

Applications of automation tests differ from manual tests unlike manual tests, automation tests are not suitable for all areas. The main uses of automation tests are regression tests, data-driven tests, smoke tests, static & repetitive tests, load and performance tests [5]. These test types are based on measurable data. Automation testing is a good tool for testing functional and non-functional test types. The use of automation test tools has improved greatly. According to International Software Testing Qualifications Board yearly test reports published worldwide, test automation in 2015-2016 was 58.5% [6], it increased to 64.4% in 2017-2018 [7]. This indicates significance of automation testing tools in future and thus motivates to further explore, analyze and compare further insight in this direction.

However, contrary to many views, automation tools need manual testing. Testing in the software development process is a high cost factor. In order to minimize these costs, test automation has been presented as an effective solution. Automation tests require manual test runs [8]. First, the manual operation of the subject to be automated should be made and then the situations to be automated should be decided. For this reason, manual tests are part of automation tests and cannot be separated. On this issue, the following that Instead of increasing human resources during the test, improving the level of test automation and risk reduction offers an approach to assist manual testing [3]. One of the most important issues for automation is to choose the automation testing tool and the appropriate framework. Thanks to the research and preliminary studies on these issues, a more successful automation process can be managed. On this topic, assessment of test automation tools is in fact a long and demanding process and much research is required during the evaluation phase [9].

The main aim of the study is to provide comparative view of important and popular test automation tools. It is aimed to draw a general framework about automation and automation tools that should be dealt with in the test processes and to shed light on future studies. Another important step in the automation testing process is the choice of testing tool. A number of information and inputs to be considered during these elections are provided in this study. Therefore, this paper will contribute towards body of knowledge in this direction and invaluable for both academic researchers and software and information technology professionals. The structure of the paper is organized as follows: Section 2 presents the related work. Section 3 outlines an automation testing frameworks. Section 4 describes briefly automation test tools. Section 5 provides discussion regarding these automation tools their features, similarity, unique features and usefulness. Section 6 is the conclusion, limitations and future research directions.

2. RELATED WORKS

There are many studies in the literature of automation test tools. There are studies that will be used especially in the selection phase of test automation tools and have certain information transfer. The dynamics within the scope of the project play an important role in the decision making process of the test tools. After healthy observations and decisions, these tests will be more efficient [10]. Various web automation testing tools and concluded Selenium is the best available automation tool for web applications [11]. There are well known and popular three automation tools Selenium free source, HP Quick test professional (QTP) and Test Complete [12]. A comparative study of automated tools observed Test Complete has easy to use user-interface and efficient playback [12]. To find best tool in selenium suite they compared it with some other tools for similar tasks and also evaluated performance on the basis of related metrics [11]. Using Selenium to automatized functional test can reduce costs. For selenium, the scenario increased its writing efforts by approximately 15% from QTP [13].

Selenium should be preferred if you do not want to spend money on the test tool [14]. Selenium is a good option when it comes to web applications [15]. Because it gives a tester more flexibility to test various complex scenarios. According to Islam, Selenium provides great advantages in terms of its many properties. Some of these features include the ability to integrate with various frameworks, the number of different programming languages, and free cost [16]. A comparative study of software test automation tools was done and recommended some tools that are mostly used in the market for automation testing. These tools include Selenium, Ranorex, Test Complete, Rest UI and more [17]. Selenium is one of the precious automation tools for different specifications. Some of these features include advanced features recording, data driven testing and ease of learning, enhanced support for other application integration [18]. If you want to be successful in tests, you should do good analytical skills, programming skills and knowledge of test tools [19]. We could not find any study which has compared many automation testing tools on twenty attributes in comprehensive manner. Therefore, this work further advances and contribute in this direction.

3. AUTOMATION TESTING FRAMEWORKS

There are various types of testing frameworks for test automation tools. Some of these methods are briefly explained in this section.

3.1. Modular Testing Framework

The modular testing framework is based on Object-Oriented Programming concepts. This framework separates the entire application under test into a set of logical and isolated modules. A separate and independent test script is required for each sub-module. Therefore, when these test scripts put together, they generate a

larger test script that represents multiple modules. An abstraction layer separates these modules so that changes to portions of application do not affect them [20].

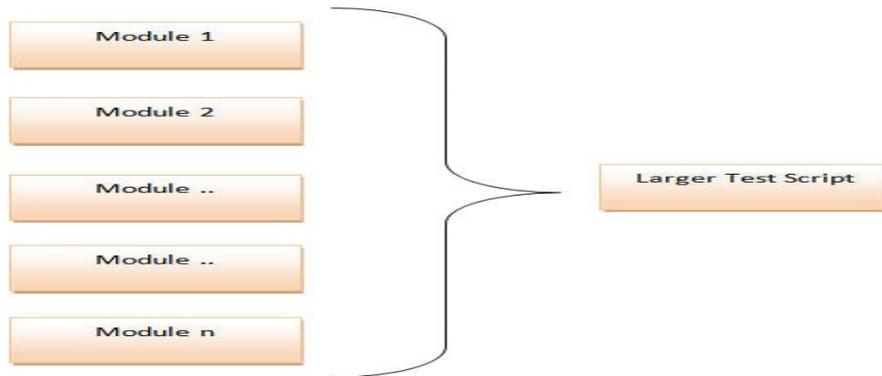


Fig. 1. Modular Testing Framework Schema [20]

3.2. Data Driven Testing Framework

In the application tests, the same functions can be tested with different data sets. Therefore, it is not a reasonable option to give test data within the automation code. Retention of data in external databases is a more logical and effective method. This framework provides a distinction between these types of data. It creates isolated test data by keeping the data in different files. Data is traditionally stored in “Key / Value” pairs [20].

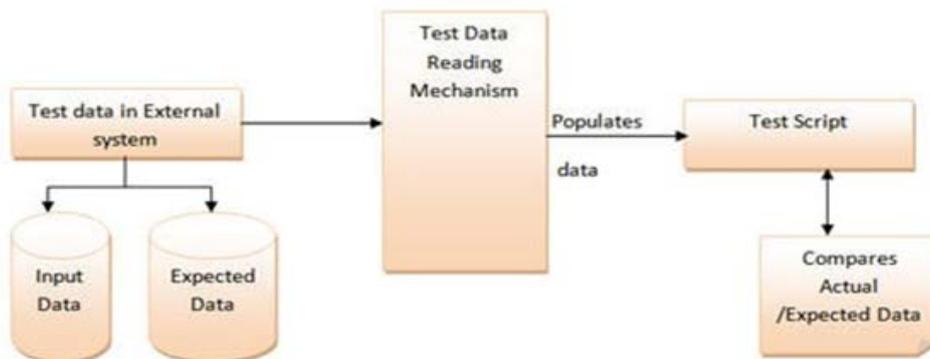


Fig. 2. Data Driven Testing Framework Schema [20]

3.3. Keyword Driven Testing Framework

The keyword-based test framework separates test data from scripts and also allows it to store a specific set of code from the test script to an external data. In terms of keywords are determined in this code set, and so the framework is named this way. Test Data and keywords kept in a table-like structure. Keywords and test data must be assets independent of the automation tool used [20].

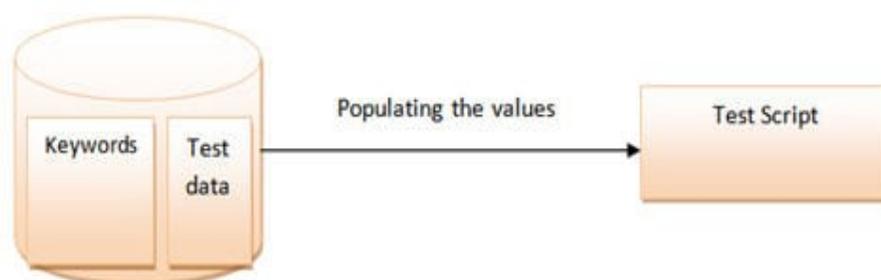


Fig. 3. Keyword Driven Testing Framework Schema [20]

3.4. Hybrid Testing Framework

The Hybrid Test Frame is a combination of multiple frames. The best thing about such installation is that it takes advantage of the different frameworks [20].

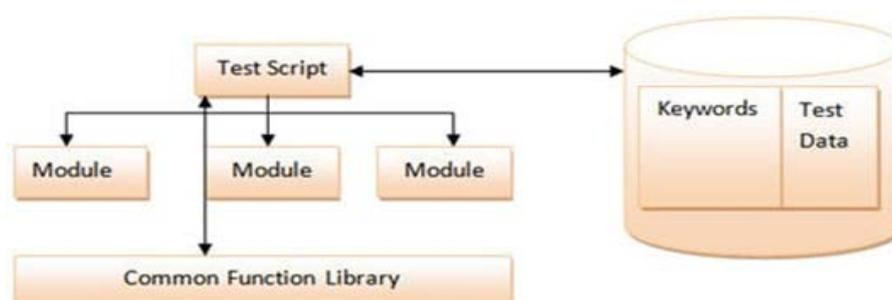


Fig. 4. Hybrid Testing Framework Schema [20]

3.5. Behavior Driven Testing Framework

The Behavior Oriented Development framework allows automation in a readable and understandable format for testers. Such frameworks do not require the user to know the programming language [20].

4. AUTOMATION TESTING TOOLS

This section provides brief information about various test automation tools available. Their comparative view on important attributes is provided in Table 1. Automation Testing Tool Features.

(1) **Selenium** has multiple test contents. These are Selenium Grid, Selenium IDE, Selenium 1 (Selenium RC or Remote Control) and Selenium 2 (Selenium WebDriver). They offer different possibilities in terms of their use. Selenium also provides multi-browser support. Some of these browsers are Chrome, Safari, Firefox and Internet Explorer. In addition, it provides support for multiple operating systems and different programming languages. Examples of programming languages include Java, Python, C # and JavaScript [21].

(2) **Protractor** also provides WebDriver feature like other platforms. In addition, multi-browser support is one of the important features. Chrome, Firefox, Safari, IE and Opera browsers are among the supported browsers [22]. Protractor is a wrapper around WebDriverJS and supports behavior-driven development frameworks. Protractor is a Node.js program that supports test frameworks [23].

(3) **Unified Functional Testing** is another automation test tool. In addition, the most important component of this tool is AI-based object recognition. There is also integration with Continuous Integration. This program is based on Windows and the software language is VBScript [24].

(4) **Appium** is a tool for mobile testing. IOS and Android support is available. In addition, using WindowsAPDriver, the windows operating system enables automation within mobile devices. There are client libraries in Java, Ruby, Python, PHP, JavaScript, and C#, which support Appium's extensions to the WebDriver protocol [25]. In Addition to these, Appium can provide REST API testing.

(5) **Test Complete**. Regardless of any code, Record and Playback or keyword-driven tests can be good methods for Automation User Interface tests. This feature ensures that the tests are repeated once after recording [26]. Test Complete supports different program languages. However, the operating system is running on windows.

(6) **The Cucumber** tool supports different languages such as Java.net and Ruby. This tool offers a different approach to the method of application and the language used. Test scenarios can be entered in plain text in English. Therefore, it does not require code information and provides ease of use. Unlike most test tools, it can help to E2E test scenarios. In addition, Cucumber can reuse the code with the code infrastructure it provides [27].

(7) **Ranorex Studio** allows end-to-end testing of tests on different devices. These devices can be desktop, web and mobile. Tests are automated on the desktop. It can then be run on native or virtual iOS or Android mobile devices or simulators, emulators. This tool also enables parallel testing. It also provides useful features from different angles thanks to the support of different browsers [28].

(8) **Watir** supports different browsers. Some of them are Chrome, Firefox, Internet Explorer, Safari, Edge [29]. Watir is an open-source platform. It used Ruby libraries in web based programs [26].

(9) **IBM Rational Functional Tester** is a self-test tool. This tool helps us automate functional, regression, GUI, and data-based testing. It supports many applications such as web-based, terminal emulator-based applications [30].

(10) **Tricentis Tosca**, accelerates testing with a script-less, no-code approach for end-to-end test automation. Tosca provides different testing types such as Risk-Based, API, Packaged App and SAP [31].

(11) **Telerik Test Studio** is a convenient tool to address functional, load and API testing needs with automation tool. Automated tests can be created for the latest technologies. In addition, mobile tests can be easily created [32].

(12) **Katalon Studio** uses Apache Groovy. Katalon Studio supports non-GUI mode for CI/CD integration. Katalon Studio does not support distributed testing. Katalon Studio only supports Web, Mobile, and API automation testing. Katalon provides Katalon Studio and Recorder for automation test. Katalon provides to exports to C#, Java, Ruby, Python, Groovy, or Robot Framework [33].

(13) **Linux Desktop Testing Project** supports different programming languages like many products. Some of them are Python, Java, Ruby, Perl. In addition, it supports operating systems such as Linux, Windows and MAC. This test tool supports OpenSuSE, OpenSolaris, Ubuntu, Debian, GNU / Linux, Fedora Core and FreeBSD [34].

(14) **Serenity** works with the requirements which need to implement. These are often expressed as user stories with acceptance criteria that help clarify the requirements. It is these acceptance criteria that we automate with Serenity. It tools can provide the report on test results and reports on functional test coverage [35].

(15) **LeanFT** supports many programming language testing technologies. LeanFT provides object identification to improve and accelerate test development. LeanFT can help create and run tests on Mac, Linux, or Windows platforms [36].

(16) **PhantomJS** is a test tool is scriptable with JavaScript and provides swift and native support for several web standards. It has the ability to work on different platforms such as Windows, Mac OS, Linux and FreeBSD. It is compatible with PhantomJS CI system and provides trouble-free installation. Functional tests can be performed on WebDriver [37]. PhantomJS is a popular tool to run unit tests. It can perform many different tests and the user can be presented with results on the command line [38].

(17) **Coded UI** provides to test functional testing. Coded UI generates code in VB/C# and it can be integrated with ALM story. Coded user interface with rich extensibility and intentional recording and flexible playback feature can be easily tested in many Coded UI applications. The application can be window or web based. Coded UI supports technologies such as Windows-based desktop applications, phone applications, web services and applications [39].

(18) **Sikuli** provides the automatic effect of Graphical user interface (GUI) tests. Sikuli can be used to automate Flash products. Desktop applications can automate by using Sikuli. In addition to this Sikuli is open source tool. It makes easy to automate windows application [40].

(19) **Applitools** is another software testing tool. With this tool devices and browsers tests can be done easily. At the same time, another unique feature, AI-powered cognitive vision, is made smoothly. AI-powered cognitive vision is provided through this tool. In addition, there is a seamless integration system with 3rd party CI tools [41].

(20) **TestArchitect** is another software automation tool known on the market. Features of this tool include cross-platform testing abilities, Keyword-Driven testing, Image based testing, Integration with CI /CD/DevOps and data & database testing [42].

(21) **Testim.io** is another test automation tool. These tool features include Data-Driven testing, link a CSV, Excel, or JSON file, Reference to data, CI tools, troubleshooting features. It also provides information and experience on multiple platforms with its reporting feature [43].

Table 1. Automation Testing Tool Features

Tool	a	b	c	d	e	f	g	h	i	j	k	l	m	n
(1)Selenium	✓	✓	✓			✓			✓		✓			
(2)Protractor		✓	✓						✓		✓			
(3)UFT		✓			✓				✓					
(4)Appium		✓	✓				✓				✓			
(5)Test Complete	✓				✓	✓	✓	✓						✓
(6)Cucumber		✓					✓		✓	✓	✓			
(7)Ranorex	✓			✓			✓				✓			
(8)Watir											✓			
(9)RFT		✓							✓		✓			
(10)Tricentis Tosca												✓		
(11)Telerik Test Studio		✓	✓								✓			
(12)Katalon Studio	✓	✓	✓	✓					✓		✓			
(13)Linux Desktop Testing Project		✓	✓											
(14)Serenity														
(15)LeanFT		✓	✓											
(16)Phantom JS		✓	✓						✓					
(17)CodedUI		✓	✓				✓				✓			
(18)Sikuli							✓				✓			
(19)Applitools					✓				✓					
(20)Test Architect									✓				✓	
(21)Testlm									✓					

Legend: a) Record and Reply; b) Multi Programming Language Support and Coding; c) Multi Operating System Support; d) Parallel Test Running; e) AI Based Object Recognition; f) Framework Support; g) Mobile Machine & Desktop Application Test; h) Debug Full-Page Screen Shots; i) Continuous Integrity; j) Reusability; k) REST & GUI Test; l) Risk Based Testing; m) Requirement Based Testing; n) Image Based Testing

5. DISCUSSION

During examining software testing automation tools it is revealed that there are multiple common and unique features. In order to meet the developing software testing needs and to come to the forefront from different perspectives, the automation tools in the market need to be analyzed well. During the survey of automation test tools, it has been observed that there are many unique features of

many different tools, as well as common features. Some of these features are as follows: Coding based Implementation, REST and GUI tests, Record and replay, multi programming language support and multi operating system support.

Many test automation tools offer coding based implementation as well as different test tools such as record and replay. Test automation tools such as Selenium, Protractor, Appium, Unified Functional Testing, Katalon, Phantom JS have successfully provided the coding implementation feature. However, the record and replay property is not as familiar and frequently used as the coding based implementation property. Record and replay features include Selenium, Test complete, Ranorex and Katalon. Another important feature is that REST and GUI tests can be tested at the same time. All automation tools have competence for backend and frontend tests at different points. Selenium, Katalon, Protractor, Cucumber and Telerik are the leading automation tools.

In addition to common features, many testing tools have their own distinctive features. Although some of these features are only available in a single testing tool, a few are also available in different testing tools. However, the characteristics of such an approach have an exceptional case. These tools offer different structures and different perspectives under the same feature. An important feature is the Multi Testing Framework Support. This feature is almost unique. Selenium and Test Complete automation tools successfully deliver this feature unlike other tools. Another unique feature is Risk based Testing. This feature is particularly important for testing high-risk projects. The only automation tool that can provide this feature is Tricentis Tosca. Katalon has introduced parallel test running, a unique feature, among other automation tools. According to this feature, multiple test scenarios provide parallel running. Although, it is not exactly a unique feature, another valuable feature is, AI based Object Recognition that is offered to users through Unified Functional Testing, Test Complete and Applitools. Since this feature is a common feature, it is a natural behavior to use different tools and find them with different application algorithms.

There are many test automation tools with multiple common and unique features. Selenium and Katalon test tools provide the most features and ease of use among these tools. These testing tools share a large place in the market. In addition, mobile tests are used for different purposes. Appium and Cucumber have a significant share in different Mobile tests. Ranorex and Selenium can have a high percentage of success in record and replay tests. In a project with a limited budget cost, it can be the main driving force for the selection process as a whole and can negotiate for other costs [44]. It is possible to examine many distinctive features of many testing tools. However, considering the general perspective and ease of use, it is concluded that the usage area of each testing tool is different. For example, although Selenium offers Record and replay feature, the most dominant and powerful feature is creating a different perspective in web applications. In terms of record and replay, the Ranorex tool is more prominent. In addition, the Appium automation tool is a capable tool for mobile device testing. The most common tool

for software testing automation is Selenium [45]. There are many reasons for this. These can be listed as free and scripts can be written in many languages. Many test automation tools have their strengths and weaknesses. Each of these test automation tools serves different purposes. A detailed analysis of these automation testing tools should be done before finally selecting any testing tool [46].

6. CONCLUSION

There are many known and unknown automation test tools in the market. The purpose of these tools and their existence is to completely fill the gaps in the market and offer the user a variety of different features. The aim of this study is to examine different automation test tools and to categorize the unique features of these test tools. These tools show many common points as well as different characteristics to cater different kind of users. However, the features of many tools examined unfortunately do not meet the needs of the market alone.

First of all, considering the fact that different platforms have different requirements, an Appium offers many unique features in the field of testing and these features meet the needs of the market from different perspectives. In addition, Ranorex performs record and reply very well. In this field, it goes beyond many products with its different perspectives and features in terms of usage in projects where code writing is not possible. When web applications are considered, Selenium, which is the choice of many users in the market, stands out. The main purpose of this is to provide different solutions for selenium in different areas. It provides automation for many projects with its solutions for different purposes.

As a result, it is not possible to mention the goodness of a single tool for automation test tools. Many projects have many different problems and infrastructures arising from the dynamics. Considering these problems and infrastructure situations, it is possible to talk about a few tools that offer different solutions. In this paper, these instruments were classified and examined in accordance with their assets and solutions.

REFERENCES

- [1] Myers, G. J., Sandler, C., & Badgett, T. *The art of software testing*. Hoboken, NJ: John Wiley & Sons, 2012.
- [2] Barus, A. C., Hutasoit, D. I. P., Siringoringo, J. H., & Siahaan, Y. A. White box testing tool prototype development. *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, 2015, pp.417–422. doi: 10.1109/iceei.2015.7352537
- [3] Mahajan, P., Shedge, H., & Patkar, U. Automation Testing In Software Organization. *International Journal of Computer Applications Technology and Research*, **4** (vol. 5), 2016, pp.198–201. doi: 10.7753/ijcatr0504.1004

- [4] Dike, S. *The Pragmatic Programmer: A Book Review*. 22 May 2018, Retrieved May 21, 2020, from <https://blog.thedigitalgroup.com/9-reasons-automation-testing-is-key-to-successful-software-development>
- [5] Fernandes, J. D., & Fonzo, A. D. *When to Automate Your Testing (and When Not To)*, March 2013.
- [6] Bazzana, Gualtiero, et al. *ISTQB® Worldwide Software Testing Practices Report 2015*, International Software Testing Qualifications Board, .2016.
- [7] Baron, Yan, et al. *ISTQB® Worldwide Software Testing Practices Report 2017*. International Software Testing Qualifications Board, 2018.
- [8] Ramler, R., & Wolfmaier, K. Economic perspectives in test automation. *Proceedings of the 2006 International Workshop on Automation of Software Test - AST 06*, 2006, pp.85-91. doi:10.1145/1138929.1138946
- [9] Abraham, C. *Aspire Systems - Technology Solutions: Digital Transformation*. 17 Sep 2019, Retrieved May 22, 2020, from <https://www.aspiresys.com/>
- [10] Ateşoğulları, D., & Mishra, A. White Box Test Tools: A Comparative View. *International Journal on Information Technologies & Security*, **3** (vol. 11), 2019, 79–90. ISSN 1313-8251.
- [11] Angmo, R., & Sharma, M. Performance evaluation of web based automation testing tools. *2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence)*, 2014, pp.731–735. doi: 10.1109/confluence.2014.6949287
- [12] Kaur H. and Gupta, G. Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete. *Int. Journal of Engineering Research and Applications*. Citeseer 2013, pp.1739- 1743.
- [13] Rajeevan, S., & Sathiyam, B. Comparative Study of Automated Testing Tools: Selenium and Quick Test Professional. *International Journal of Engineering and Computer Science*, **7** (vol. 3), 2014.
- [14] Bharti, T., & Dutt, E. V. Relative Review of Automated Testing Tools: (QTP) Quick Test Professional, Selenium and Test Complete. *International Journal of Computer Science Trends and Technology (IJCST)*, **6** (vol. 2), 2014, pp.110–114.
- [15] Kunte, P., & Mane, D. Automation Testing of Web Based Application with Selenium and HP UFT (QTP). *International Research Journal of Engineering and Technology (IRJET)*, **6** (vol.4), 2017, pp. 2579–2583.
- [16] Islam, N. A Comparative Study of Automated Software Testing Tools. *Culminating Projects in Computer Science and Information Technology*, 2016.

- [17] Shaukat, K., Shaukat, U., Feroz, F., Kayani, S., & Akbar, A. Taxonomy of Automated Software Testing Tools. *International Journal of Computer Science and Innovation*, **1**, 2015, pp. 7–18.
- [18] Singh, I., & Tarika, B. Comparative Analysis of Open Source Automated Software Testing Tools: Selenium, Sikuli and Watir. *International Journal of Information & Computation Technology*, **4** (vol. 15), 2014, pp. 1507–1518.
- [19] Mishra, D., Hacaloglu, T., & Mishra, A. Teaching Software Verification and Validation Course: A Case Study. *International Journal of Engineering Education*, **6** (vol. 30), 2014, pp. 1476–1485.
- [20] Umratkar, A., Hoang, T., Testingspree, Dores, E., & Gabriel. *Most Popular Test Automation Frameworks with Pros and Cons of Each – Selenium Tutorial #20*. 16 April 2020, Retrieved May 21, 2020, from <https://www.softwaretestinghelp.com/test-automation-frameworks-selenium-tutorial-20/>
- [21] *Selenium automates browsers. That's it!* Retrieved May 21, 2020, from <https://www.seleniumhq.org/>.
- [22] *End-to-end testing for AngularJS*. Retrieved May 21, 2020, from <http://www.protractortest.org/#/>.
- [23] Anuradha. *Why Use Protractor for End-to-End Testing of AngularJS Applications?* 2016, Retrieved May 21, 2020, from <http://www.tothenew.com/blog/why-use-protractor-for-end-to-end-testing-of-angularjs-applications/>
- [24] *Automate Functional Testing from UI to the API: UFT One*. Retrieved May 21, 2020, from <https://www.microfocus.com/en-us/products/unified-functional-automated-testing/overview>
- [25] *Appium*. Retrieved May 21, 2020, from <http://appium.io/>.
- [26] *Watir*. (2019, August 30). Retrieved May 21, 2020, from <https://en.wikipedia.org/wiki/Watir>
- [27] *Cucumber - Overview*. Retrieved May 21, 2020, from https://www.tutorialspoint.com/cucumber/cucumber_overview.htm
- [28] *Test Automation for GUI Testing*. Retrieved May 21, 2020, from <https://www.ranorex.com/>.
- [29] *Project, W*. Retrieved May 21, 2020, from <http://watir.com/>.
- [30] *Rational Functional Tester*. (2020, May 21). Retrieved May 21, 2020, from <https://www.ibm.com/us-en/marketplace/rational-functional-tester>

- [31] *Automated Software Testing Tools*. (2020, May 20). Retrieved May 21, 2020, from <https://www.tricentis.com/>.
- [32] *Automated Testing for Web, Mobile and Desktop Apps - Telerik Test Studio*. (a). Retrieved May 21, 2020, from <https://www.telerik.com/teststudio>.
- [33] *Features*. (2020, May 21). Retrieved May 21, 2020, from <https://docs.katalon.com/faqs/features.html#q-what-is-katalon-studios-scripting-language>
- [34] *ldtp*. Retrieved May 21, 2020, from <https://ldtp.freedesktop.org/wiki/>.
- [35] *Serenity BDD - Automated Acceptance Testing with Style*. (a). Retrieved May 21, 2020, from <http://www.thucydides.info/#/>.
- [36] *Shift-Left and Automate Agile Functional Testing: UFT Developer*. Retrieved May 21, 2020, from <https://www.microfocus.com/en-us/products/automated-continuous-testing-integration-delivery-tools/overview>
- [37] *Scriptable Headless Browser*. Retrieved May 21, 2020, from <https://phantomjs.org/>.
- [38] Scotch. *What is PhantomJS and How is it Used?* 12 Nov 2018; Retrieved May 21, 2020, from <https://scotch.io/tutorials/what-is-phantomjs-and-how-is-it-used>
- [39] *Coded UI Test Automation Framework Tutorial*. Retrieved May 21, 2020, from <https://www.guru99.com/coded-ui-test-cuit.html>
- [40] Subbrao, Kathyayani, Nguyen, K., Tim, & Javier. *Introduction to Sikuli (GUI Automation Tool) - Sikuli Tutorial Part 1*.16 April 2020; Retrieved May 21, 2020, from <https://www.softwaretestinghelp.com/sikuli-tutorial-part-1/>
- [41] *AI Powered Visual UI Testing & Monitoring*. (2020, April 30). Retrieved May 21, 2020, from <https://applitools.com/>.
- [42] User, S. *Test Automation Software*. Retrieved May 21, 2020, from <https://www.testarchitect.com/>.
- [43] *Automated Functional Testing - Software Testing Tool*. (2020, May 12). Retrieved May 21, 2020, from <https://www.testim.io/>
- [44] Raulamo-Jurvanen, P., Mäntylä, M., & Garousi, V. (2017). Choosing the Right Test Automation Tool. Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering - EASE17, 1–11. doi: 10.1145/3084226.3084252
- [45] Oliinyk, B., & Oleksiuk, V. Automation in software testing, can we automate anything we want? *2nd Student Workshop on Computer Science & Software Engineering*, 2019, pp. 224-234.

[46] Umar, M. A., & Chen, Z. A Study of Automated Software Testing: Automation Tools and Frameworks. *International Journal of Computer Science Engineering (IJCSE)*, 6 (vol. 8), Nov-Dec 2019, pp. 217-225.

Information about the authors:

Dilara Ateşoğulları – Dilara Ateşoğulları is Senior Software Test Engineer. She has especially worked on test automation. She has been involved in multiple projects on Selenium.

Alok Mishra – Dr. Alok Mishra is Professor Informatics and Digitalization, Molde University College (Specialized University in Logistics), Norway. He is also associated as Professor Software Engineering, Atilim University, Turkey. His areas of research interest are Software Engineering, Information System, Artificial Intelligence and Digitalization.

Manuscript received on 15th October 2020