

AN EMPIRICAL STUDY OF USER INTERFACE TESTING TOOLS

Elis Pelivani(1), Adrian Besimi(2), Betim Cico(3)*

^{1,2} South East European University, Tetovo, Macedonia Faculty of Contemporary Sciences and Technologies; ³Metropolitan Tirana University, Albania Faculty of Computer Science & IT
North Macedonia^{1,2}, Albania³

* Corresponding Author, e-mail: ep24253@seeu.edu.mk

Abstract: Software testing is gaining much more importance in the software development life cycle. Many companies are investing in several testing methods and especially in test automation. Nevertheless, it is quite difficult to choose the best tool or framework to use. In this study, we will do an evaluation and comparison of Katalon Studio, Selenium and UFT in a web application. This is the preliminary work to achieve some solid outcomes in order to propose and create a new general framework for test and code optimisation.

Key words: testing, automation, tool, framework.

1.INTRODUCTION

This study is part of a research in software testing tools and frameworks for the PhD that aims to present a new general framework for test optimization that can support the technical extensions of several types. A short comparison of software testing tools and frameworks is already introduced and published in MECO conference.

Software testing has become essential for the software industry. Testing a software, helps in measuring the quality of software in terms of errors found, for both, functional and non-functional characteristics such as reliability, usability, efficiency and maintainability.

With the increase of testing importance in the software industry, different tools, which perform testing, were introduced into the market. Tools with capabilities of screen navigation capture, presented a new interesting way to test applications. Automated testing tools that provided this capability, became preferred. Using automated testing tools requires people who perform testing to be more familiar with the application under test.

There are many comparative studies already done for the software testing frameworks and tools. However, we went beyond the empirical analysis of three testing tool compared in standard metrics such as in other studies. We focused in the execution time and used two important metrics number of steps and number of object. The study was intentioned to explain such results focusing in how these tools are designed to operate and offer. Some of the main topics that we have analysed and explained are the popup handling, identification of objects, database testing, reporting, capture and recording versus programming and usage of AI. The last one of course needs much deep study because new releases of the testing tools are improving the AI usage very fast, as a need of the software industry.

2. SIMILARITIES AND DIFFERENCES OF THE TESTING FRAMEWORKS

2.1. Test Frameworks

A brief review of Selenium, Katalon Studio and UFT is provided in this section by pointing out their main characteristics.

Selenium is a free, open source framework that conducts functional testing of web applications. It is running crosswise through various browsers and systems, such as Windows, Mac, and Linux. Selenium encourages developers to compile tests in a range of programming languages such as Java, PHP, C#, Python, Groovy, Ruby, and Perl. Selenium Suite consists of four critical segments: Selenium IDE, Selenium RC, WebDriver, Selenium Grid, is designed to support the automation testing [1].

Katalon Studio is an automated testing framework that provides a robust range of features to incorporate completely automated testing solutions for Web, API and Desktop and Smartphone apps. Designed on top of the open-source Selenium and Appium frameworks, Katalon Studio helps teams to easily get started with test automation, reducing the effort and skills needed to understand and incorporating these systems for automated testing needs. Katalon conducts UI, functional and regression testing [2, 3].

Unified Functional Testing (UFT) is a Windows-based programming testing framework used to test software on the desktop or the internet, providing test automation for regression and functional testing, developed by Hewlett Packard (HP). UFT uses the Visual Basic Script language to run scripts and evaluation formats are used for various artifacts and program verification controls. UFT conducts functional and regression testing [3, 4].

2.2. Capture and recording versus programming

As easy as it gets, Katalon Studio is a tool you can use for implementing simple test cases, when the programming knowledge of the tester is limited. Katalon Studio uses three ways to test Record and Replay, Manual Mode and lastly the Script Mode. In the Record & Replay method, the tester records all the steps and actions performed while using the application and the functionalities being tested. These steps are saved

in order to be run when needed or when a new change has been done to the application/functionalities. Same as other test cases, this type of tests can be updated by changing the recorded steps as the test idea changes with the time. Scripting mode on the other hand does require a little of programming knowledge, as the tester needs to write Groovy code to create tests. Usually this is done to customize an existing (recorded) test and to create custom keywords. The later are just advanced functions [4, 5]. In contrary with Katalon and UFT, each tester is required to have advanced programming knowledge to build the framework, create the structure of the testing project, and implement functions and scenarios, Table 1.

Table 1. Differences and similarities between Katalon Studio, UFT and Selenium [5, 6, 7, 8]

Features	Katalon Studio	Selenium	UFT
<i>Test deployment platform</i>	<i>Cross – platform</i>	<i>Cross – platform</i>	<i>Windows</i>
<i>Application under test</i>	<i>Web and mobile apps</i>	<i>Web apps</i>	<i>Windows desktop, Web, mobile</i>
<i>Scripting languages</i>	<i>Groovy</i>	<i>Java, C#, Perl, Python, Javascript, Ruby, PHP</i>	<i>VBScript</i>
<i>Programming skills</i>	<i>Not requires, Recommended for advance test scripts</i>	<i>Advanced skills needed to integrate various tools</i>	<i>Not required, Recommended for advanced test scripts</i>
<i>Learning curves</i>	<i>Medium</i>	<i>High</i>	<i>Medium</i>
<i>Ease of installation and use</i>	<i>Easy to setup and run</i>	<i>Require installing and integration various tools</i>	<i>Easy to setup and run</i>
<i>Script creation time</i>	<i>Quick</i>	<i>Slow</i>	<i>Quick</i>
<i>Object storage and maintenance</i>	<i>Build-in Object repository, Xpath object identification</i>	<i>Xpath, UI Mpas</i>	<i>Built-in object repository, smart object detection and correction</i>
<i>Image-based testing</i>	<i>Built-in support</i>	<i>Require installing additional libraries</i>	<i>Built-in support, image-based object recognition</i>
<i>Continues integrations</i>	<i>Popular CI tools (e.g. Jenkins, Teamcity)</i>	<i>Various CI tools (e.g. Jenkins, Cruise Control)</i>	<i>Various CI tools (e.g. Jenkins HP Quality Center)</i>
<i>Product support</i>	<i>Ticketing support community</i>	<i>Open source community</i>	<i>Dedicated stuff, community</i>
<i>License type</i>	<i>Freeware</i>	<i>Open source</i>	<i>Proprietary</i>
<i>Cost</i>	<i>Free</i>	<i>Free</i>	<i>License and maintenance fees</i>

UFT or Unified Functional Testing is a commercial automated functional testing tool. Test generation is one among the foremost important feature of UTF. When navigating through the application, UTF/QTP records each step and creates the test script. Once a test is generated, it is possible to form some amendments within the test script. The latest version of UTF includes Artificial Intelligence to simplify and improve test creation, execution, and maintenance. AI is used for object recognition, mockup identification, screen recording, text matching, and image identification. UFT uses AI to identify objects visually, based on a wide variety of images, context, and in some cases text.

2.3. Popup handling

Selenium works for WebElements, which are part of HTML inside a browser, but it will fail to handle special components that are not part of HTML such as Windows popups, Modal dialogs or Browser popups and alerts.

While Katalon Studio and UFT are more suitable to handle these kind of elements. Both of them provides capabilities to handle Windows controls, modal and browser dialogs.

2.4. Identification of objects

The most important aspect of automating applications is the identification of objects. For object identification and understanding, UFT provides a very important tool called Object Spy. Object spy helps to add objects in the Object Repository. The Object Spy helps us to reach all the native properties and operations of any object in an open application, as well as the test object hierarchy, description properties, and operations that UFT uses to represent that object.

Katalon Studio allows users to locate Test objects in two different modes. Basic mode is recommended to manual testers who just started their automation experience. It is possible using the Web Object Spy function. With Basic mode, some XPath's are automatically generated for the same object. For advanced testers who wish to input selectors manually, there is the possibility to create the Object manually and to select between CSS or XPath mode. After the objects are created, they can be saved in the Objects Repository in the desired folder.

Selenium uses locators instead of Object Spy. An HTML DOM element can be selected by a locator and then can be manipulated in the way that the tester needs. Locators in Selenium come into action when we perform designated actions in a defined test after the Selenium WebDriver is initialized and loaded the webpage to be tested. There is not possible to find automatically an element selector, but you have to inspect the element inside the DOM structure and find a unique way to trigger it. Common ways to locate a web element are by using the id, class name, or xpath Selenium-Java.

2.5. Database Testing

For database testing, we need some basic methods to be implemented like Create Connection, Execute Query, and Close Connection. Katalon Studio offers the possibility to create custom keywords to address specific needs. All basic database methods can be created as custom keywords and then can be called on test case steps when needed [9]. While on the Selenium framework, using Java as a programming language, database testing requires configurations to be done. As a java project, it needs to import the packages, to register the JDBC driver and to implement the database methods. Similar to methods are created as keywords on Katalon Studio.

UFT does not provide any built-in support to connect to databases, however, using VBScript testers will be able to connect and interact with databases using ADODB connection objects. Considering that the three tools studied in this paper are browser automation tools, they are not recommended for database testing though it is possible. There are better tools for this purpose.

In a large systems where is needed both web testing and database testing would be required to have different testing tools. In order to tackle this limitation a customized, which optimize the database testing for web applications, would be a very acceptable solution.

3.EXPERIMENTAL RESULTS

3.1. Environment and testing frameworks versions

We have selected a real life web application that is used from the employees of a company mainly to select the siting positions in the offices, selects the days off and the home office.

Same computer type used for Selenium+ Cucumber, UFT, and Katalon Studio executions, Table 2. Test cases in Selenium can be created in Java, C#, Perl, Python, JavaScript, Ruby, PHP. In this research, we have created our test cases by using Java in Selenium, Groovy in Katalon and VBasic in UFT.

Table 2. The simulated test environment

	<i>Selenium</i>	<i>Katalon Studio</i>	<i>UFT</i>
<i>System Model</i>	<i>Dell Latitude 5580</i>	<i>Dell Latitude 5580</i>	<i>Dell Latitude 5580</i>
<i>RAM</i>	<i>16.0 GB</i>	<i>16.0 GB</i>	<i>16.0 GB</i>
<i>Operating System</i>	<i>Windows 10 64bit</i>	<i>Windows 10 64bit</i>	<i>Windows 10 64bit</i>
<i>Tool version</i>	<i>3.141.59</i>	<i>Katalon 7.9.1</i>	<i>UFT 15.0.2</i>
<i>Browser used</i>	<i>Chrome 80.0</i>	<i>Chrome 80.0</i>	<i>Chrome 80.0</i>

3.2. Our Experiments

The same test cases are implemented and executed for Selenium, UFT and Katalon Studio Tool. There are 10 test cases executed, which have around 240 steps. During this simulation we have used only UI testing.

One of the main reasons why using automation testing is the reusability of code or objects. In the above table are displayed the numbers for objects used in each test case. The number of total created objects is smaller than the sum of objects used in each test. This proves that a lot of test objects or WebElements are reused. Table 3 shows all the objects and steps that are created for each test used for this paper. The average usage of an element is calculated with the below formula:

$$\text{Average usage of an element} = \frac{\text{Total No of Steps}}{\text{Total Object Created}}$$

Table 3. Number of steps and objects used on each test case

	Selenium		Katalon Studio		UFT	
	No. of Steps	No. of Objects	No. of Steps	No. of Objects	No. of Steps	No. of Objects
<i>T1-CheckLoginFunction</i>	15	10	22	12	16	10
<i>T2-EditProfile</i>	25	17	46	25	34	13
<i>T3-InOfficeMode</i>	24	23	46	32	35	15
<i>T4-HomeOfficeMode</i>	17	18	35	23	32	12
<i>T5-OnLeaveMode</i>	17	18	35	23	30	12
<i>T6-TravelMode</i>	17	18	35	22	32	12
<i>T7-EmployeesReports</i>	21	21	37	23	25	17
<i>T8-DailyBookings</i>	11	11	19	12	13	12
<i>T9-DeleteBookings</i>	8	10	19	12	16	10
<i>T10-UpdatePasswords</i>	23	21	36	22	25	22
<i>Total</i>	178	93	330	113	258	135
<i>Average usage of an element</i>	~18	1.80	33	2.92	25.8	1.91

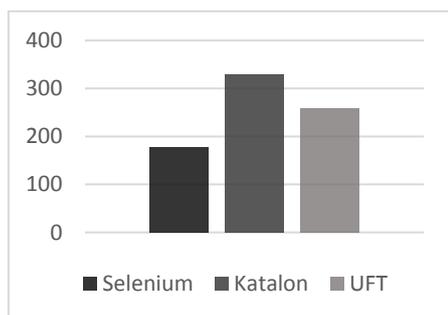


Figure 1. Total number of steps used on each tool

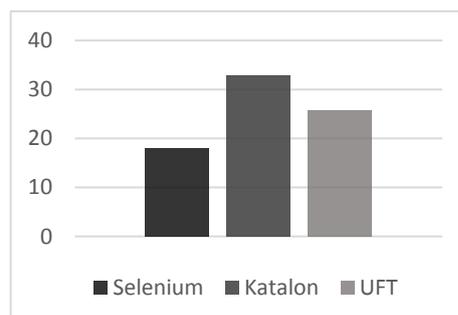


Figure 2. Average number of steps used on each tool.

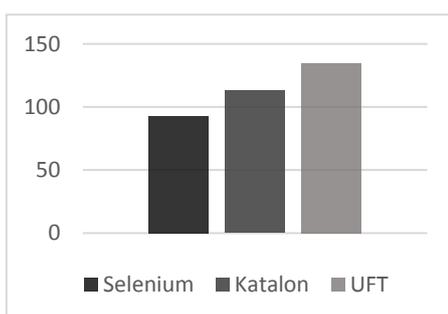


Figure 3. Total number of object used on each tool.

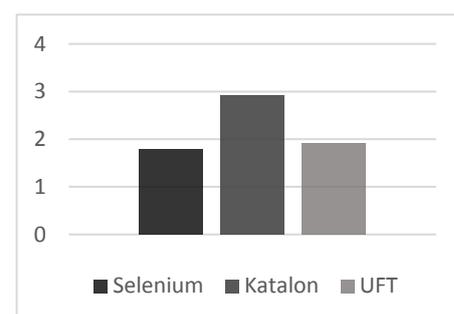


Figure 4. Average number of objects used on each tool

3.3. Running and Execution time

Obviously, the execution time for Katalon Studio is higher than Selenium and UFT, Figure 5.

The main reason for that is the programming language used. Katalon Studio uses Groovy, a language built on top of Java, and must load many libraries for parsing test data, test objects, logging while for Selenium framework is used Java itself and for UFT VBScript. UFT is slower than Selenium because VBScript cannot handle pointers in a direct way. This leads to many CPU cycles and requires more time for processing, while Java can handle pointers in a faster way.

UFT cannot run more than one test case at a time. We need extra configuration by creating a test chain with ALM (Application Lifecycle Management). ALM is an application, which can store test cases, manage version control, and create test chains, test suits.

In Selenium, we have to create a TestRunner class that manage the running of test cases. There are specified where our scenarios are defined, where is located the implementation and which scenario we have to run. Using Katalon Studio we have the possibility to run the tests one by one, by clicking Run button and choosing the desired browser.

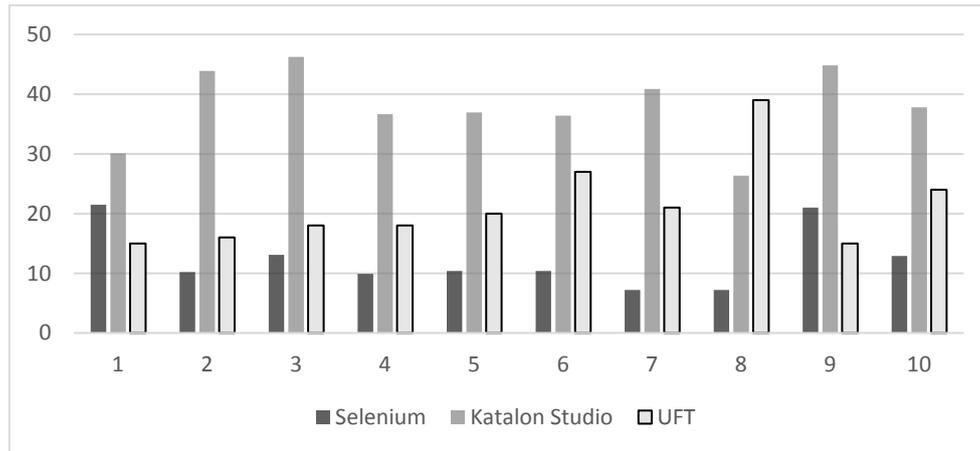


Figure 5. Execution time of Katalon studio, UFT and Selenium

Running all tests is possible by using a suite file under the Test Suites package. On the Suite file, we can add all test cases and activate the checkbox for each test we want to run.

3.4. Reporting

A detailed, user-friendly, and easy to read reports can be exported into HTML, CSV, and PDF files in Katalon Studio. These reports can be easily shared with team members or managers anytime. There are screenshots done automatically for each case, failures or success without extra configuration, Figure 6.

Suite 1			
Execution Environment			
Host name	Host Name		
OS	Windows 10 64bit		
Katalon version	7.9.1.208		
Browser	Chrome 80.0.3987.132		
Summary			
ID	Test Suites/Suite 1		
Description			
Total	10	Failed	0
Passed	10	Incomplete	0
Error	0		
Start	2021-04-01 01:21:20	End	2021-04-01 01:27:55
Elapsed	6m - 35.805s		
#	ID	Description	Status
1	Test Cases/CheckLoginFunction		PASSED
2	Test Cases/EditProfile		PASSED
3	Test Cases/InOfficeMode		PASSED
4	Test Cases/HomeOfficeMode		PASSED
5	Test Cases/OnLeaveMode		PASSED
6	Test Cases/TravelMode		PASSED
7	Test Cases/EmployeesReports		PASSED
8	Test Cases/DailyBookings		PASSED
9	Test Cases/DeleteBookings		PASSED
10	Test Cases/Update password		PASSED

Figure 6. The report generated from Katalon studio

The same user-friendly interface for test reporting is provided in UFT. For each test case, there is a report generated and screenshots saved in the repository for each step by default, Figure 7.

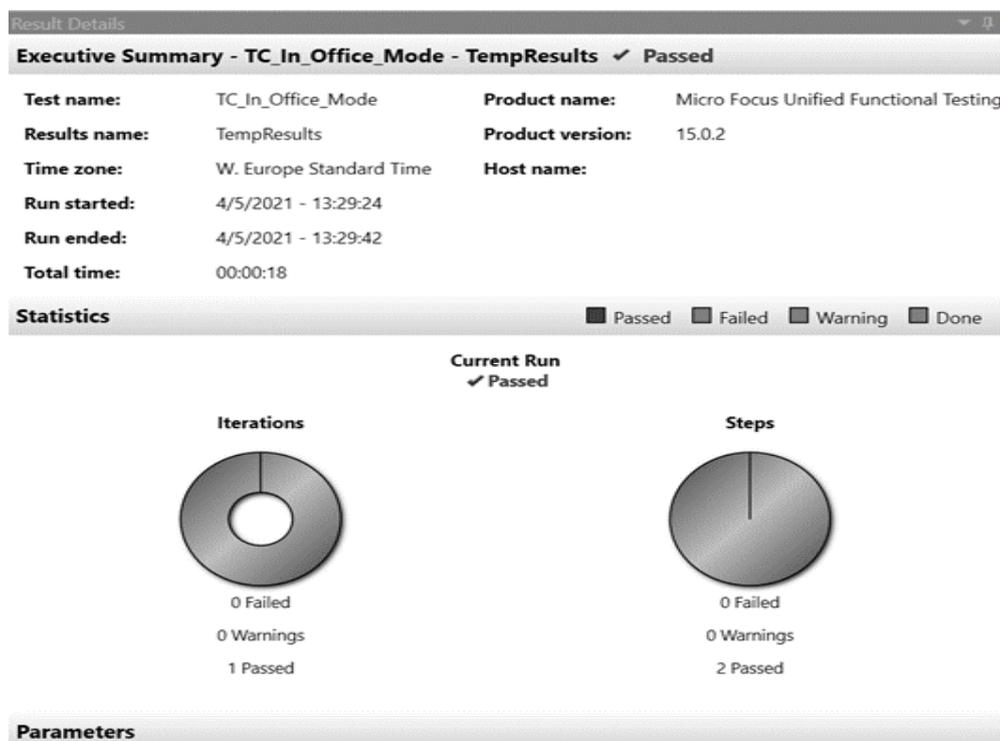


Figure 7. The report generated by UFT

Selenium does not have reporting tools; extra configurations are needed to be done based on the framework used like TestNG, Cucumber, and so on, which have some simple options of reporting templates [10, 11].

Reporting can be considered as one of the advantages of Katalon Studio and UFT over Selenium framework. Katalon and UFT, as shown in the previous figures, give more information regarding the test cases, machines and browsers that are used. When a test fails, we can see through report generation and we can easily document those reports. Meanwhile on Selenium we have to go through logs to check if everything worked and if something failed, which will take more time than in two other tools [12, 13].

4. USAGE OF AI IN TESTING TOOLS AND CONCLUSION

UFT One is an Artificial Intelligence Feature that is used to identify objects in the same way that we would. This feature helps you to run the same test on different platforms and versions, despite objects' implementation. Objects can be identified

visually, based on a wide variety of images, context, and text. According to UFT documentation, while using AI-based object identification the test scripts are more intuitive, technology agnostic, and easier to maintain [13, 14]. This feature seems very helpful and easy to use, but practically it is not. It needs deep knowledge in Visual Basic. All objects that are recognized by AI need to be included not only with drag and drop but with some scripts before we run the test. It is not easy for everyone to use this feature and to maintain tests that are created with AI-based object recognition. Another thing that makes this feature not so practical for usage is that it only works on Internet Explorer. This feature does not know other browsers and you have to create the script from scratch [15].

Artificial Intelligence features can be used on Katalon by installing two plugins, Auto Healing Smart Path and Appitools. Auto Healing Path is mostly used for UI testing for finding xpath while the automation suite is being executed. In this way the UI testing is easier, with higher performance is provided and accuracy [16, 17, 18]. This plugin helps us in cases where the xpath is changed and we do not need to change it manually, AI provides the correct xpath and the test does not fail. Appitools uses artificial Intelligence for visual comparison. This plugin offers features, which can compare and verify the layout of the application in different screen resolutions and sizes. These tools help us to have smarter and efficient test cases, reduce the time of implementation and maintenance of the test suite.

In contrary to Katalon Studio and UFT, Selenium does not have any features related to Artificial Intelligence. AI can only be integrated via open-source libraries.

In this paper, we did an empirical study of three widely used test automation tools, Selenium, UTF and Katalon Studio, based on different factors. We concluded that it is very difficult or not even possible to have a rank of the software testing tools only based in comparison and some test cases results. There are several aspects to be considered when we select a testing tool, such as: the need for testing, the application scope, the required efficiency and the available budget. The features and the limitations of each tool have to be analyzed before the final decision what to use.

REFERENCES

- [1]. Harpreet Kaur, Gagn Gupta. Comparative study of automated testing tools: Selenium, quick test professional and test complete. *International Journal of Engineering Research and Applications*, ISSN: 2248-9622, Vol. 3, No. 5, Sep-Oct 2013, pp.1739-1743.
- [2]. Duaa Mohammad, Sajedah Al-Momani, Yahya Tashtoush and Mohammad Alsmirat. A comparative analysis of quality assurance automated testing tools for windows mobile applications. *IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 0414-0419, doi: 10.1109/CCWC.2019.8666463.

- [3]. Vinita Malik, Mamta Gahlan. Comparative study of automated web testing tools. *International Journal of Latest Trends in Engineering and Technology*, ISSN: 2278-621x, Vol.6, No.3, 2016, pp.367-374, <https://www.ijltet.org/journal/55.pdf>
- [4]. Anjum, H., Babar, M.I., Jehanzeb, M., Khan, M., Iqbal, S., Sultana, S., Shahid, Z., Zeshan, F., Bhatti, S. A comparative analysis of quality assurance of mobile applications using automated testing tools. *International Journal of Advanced Computer Science and Applications*. Vol. 8, No.7, 2017, pp.249-255.
- [5]. Ashish Bhargava. Designing and implementing test automation frameworks with QTP: Learn how to design and implement a test automation framework block by block. *Packt Publishing*, ISBN 9781782171027, Vol.1, 2013, pp 0-160.
- [6]. Monier, M., El-mahdy, M. Evaluation of automated web testing tools. *International Journal of Computer Applications Technology and Research*. 4, 2015, pp.405-408.
- [7]. Daniel Asfaw. Benefits of automated testing over manual testing. *International Journal of Innovative Research in Information Security*, ISSN: 2349-7017, Vol.2, 2015, pp 5-13.
- [8]. Divya Kumar, Krishna Kumar Mishra. The impacts of test automation on software's cost, quality and time to market. *7th Int'l Conference on Communication, Computing and Virtualization*, Procedia Computer Science, Vol.79, 2016, pp. 8-15.
- [9]. Ateşoğulları, D. Mishra, A. White Box Test Tools: A comparative view. *International Journal of Information Technologies and Security*. Vol. 11, No.3, 2019, pp 79-90.
- [10]. Mojtaba Shahin, Muhammad Ali Babar, Liming Zhu. Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices, *IEEE Access*, vol. 5, 2017, pp. 3909-3943.
- [11]. Selenium umbrella, *Selenium Documentation: Test Automation for Web Applications*, www.seleniumhq.org, last update 11 November 2021.
- [12]. Dzung Ngo. *Katalon Studio and Applitools: When Codeless Meets AI*, <https://applitools.com/>, last update July 16, 2020.
- [13]. AltexSoft Team, *The Good and the Bad of Katalon Studio Automation Testing Tool*. <https://www.altexsoft.com/>, last update 15 Jun 2021
- [14]. Microfocus Team, *UFT One Help Center*, version 15.0-15.0.2, <https://admhelp.microfocus.com/>, last update 5 September 2021.
- [15]. Katalon Team, *A Comparison of Automated Testing Tools*, <https://www.katalon.com>, last update 24 January 2017

[16]. Nazia Islam. A comparative study of automated software testing tools. *Culminating Projects in Computer Science and Information Technology*. The Repository at St. Cloud University, Vol. 12, 2016, pp 0-92.

[17]. Prashant Malhotra. Test automation tool comparison – HP UFT/QTP vs. Selenium, Aspiresys. Last update 25 November 2013.

[18]. Dilara Ateşoğulları, Alok Mishra. Automation testing tools: A comparative view. *International Journal on Information Technologies and Security*, Vol. 12, No. 4, 2020, pp. 63-76.

Information about the authors:

Elis Pelivani – PhD student at South East European University, Faculty of Mediterranean University of Albania, areas of research- Software testing, Artificial intelligence, Software defined Networking.

Adrian Besimi – Vice-rector for Academic Planning and Digitalization SEEU; Associate Professor, South East European University at Faculty of Contemporary Sciences and Technologies, research interest in fields of B2B, Data Mining, Web Services.

Betim Cico – full professor in Metropolitan, Tirana University, Albania; research interest in fields of Cloud computing, Digital System Design, Digital Image Processing, Advance Computer Architecture, Internet of Things, FPGA and software Engineering. He is author of many research papers, projects and supervisor of several PhD students

Manuscript received on 15 November 2021