A Study of Automated Software Testing: Automation Tools and Frameworks

Mubarak Albarka Umar [1*], Chen Zhanfang [2]
[1,2]School of Computer Science and Technology,
Changchun University of Science and Technology, 7186 Weixing Road, Jilin, China
Corresponding Author: [*1]2018300037@mails.cust.edu.cn, [2]chenzhanfang@cust.edu.cn

Abstract - The growing demand for delivering quality software faster "Quality at Speed" requires faster and successful execution of software testing to ensure its standard. Utilizing appropriate testing method(s) and right test automation tools/framework are two defining factors for a successful and effective software testing project. Using one testing method will not be sufficient to test software and ensure its standard, a combination of some appropriate testing techniques is often required. Likewise, is no one tool that can satisfy all automated testing needs which makes finding the right tool combination difficult. Knowing the various testing methods and tools/frameworks is the first step towards achieving a successful and efficient software testing. This article presents a comprehensive study of test automation tools and frameworks. Firstly, automated testing and their categories were explained, followed by an explanation of the various test automation frameworks. Finally, a brief explanation and comparison of some of the most commonly used automation tools was presented.

Keywords*: Software Testing, Software Automated Testing, Automation Tools Categories, Test Automation Tools, Test Automation Frameworks*

I. INTRODUCTION

Software programs has become an integral part of our everyday life as it potentially touches millions of people in various domain of life [1], this calls for safe and reliable software. Unfortunately, humans are prone to error, therefore the fundamental facts of humans' core involvement in software development makes errors an inevitable inclusion in software [2]. Software errors (bugs) can negatively affect the live operation and even cause death [3]. It is important to treat such errors early in the development phase because they get costlier with progress [4]. For example; Jones, in his survey [5], shows that $500 billion is lost annually due to poor software quality, furthermore, a report released by the National Institute of Standards and Technology (NIST) shows that Software bugs are costing the USA economy an estimated $59.5 billion annually, and about a third of the cost ($22.5 billion) could be reduced through improvements in testing, although it won't eliminate all the bugs [6]. The eminent effects of software bugs cannot be overestimated and hence, the need for software to be tested before delivered.

Software testing is defined as the process of evaluating a software program with the intent of finding fault or errors. Software testing is also done to; make sure that software performs its intended purpose correctly [1], access, achieve and preserve the quality of a software [2], and thereby verify that the software is fit for use [7]. Software testing is an integral phase in Software Development Life Cycle (SDLC) process [8], testing utilizes around 50% of software projects' development time and effort [1], [2], [7]. In SDLC, software is incomplete until it has passed testing [9]. The overall purpose of testing is not to demonstrate that the system is free of errors but to give confidence that the system is working well before installation. An error-free system shows that either no testing or poor testing was performed on the software system [2].

Software testing can be manual or automated. Manual Testing does not require knowledge of any testing tool, it requires lots of effort and more people. Automating manual testing is no different from a programmer using a coding language to write programs to automate any manual process [10]. Automated testing involves the use of automation tools and/or with frameworks in executing tests. It requires knowledge of automation tools and, sometimes, programming skills. Automated testing increases testing accuracy and saves effort and time of tester compared to manual testing [11], which is not only a costly and time-consuming exercise, but it is also prone to error.

Defining factors for successful and efficient software testing projects are: (a) selecting and using an appropriate testing method, and (b) choosing and using the right test automation tool and/or framework [12]. Software testing methods are the various strategies or approaches used to test an application to ensure it behaves and looks as expected. These encompass everything from front to back-end testing, including unit and system testing. Umar [13] provides a detailed explanation on software testing methods, this paper focuses on software test automation, specifically, on automated software testing, its various tools, and their categories as well as testing frameworks that can be used to achieve effective, and successful testing project.

The rest of this paper is organized as follows: Section I is the Introduction, followed by Automated Software Testing and their categories in Section II, then Section III presents Test Automation Frameworks and their types. In Section IV, an explanation and comparison of some popular test automation tools were provided and the last Section (V) Concludes the paper.

II. AUTOMATED SOFTWARE TESTING

Automated testing is a process using software separate from the software under test to control the execution of tests and the comparison of actual outcomes with the expected outcomes [14]. Automation tools are used to automate certain sections of manual testing but not all [11]. Automated testing generally saves time, the tester can efficiently run a large number of tests in a short period and so important and repetitive tasks, as well as testing that would be difficult to do manually, can be automated. Besides saving time, automation testing saves money and effort, increases the quality of the testing tasks [12] and also helps in improving software accuracy. Test Automation requires a skilled tester with knowledge of the automation tools and the software being tested to set up the test cases and perform the testing [10].

Table 1: Advantages and Disadvantages of Test Automation

Advantages	**Disadvantages**
Improves accuracy and quick finding of bugs compared to manual testing	Choosing the right tool requires considerable effort, time, and evolution plan.
Saves time and effort by making testing more efficient	Requires knowledge of the testing tool.
Increases test coverage because multiple testing tools can be used at once allowing for parallel testing of different test scenarios	Cost of buying the testing tool and, in the case of playback methods, test maintenance is a bit expensive
Automation test script is repeatable	Proficiency is required to write the automation test scripts.

A. Automation Tools Categories

Software testing automation tools can be divided into different categories as follows: Unit Testing Tools, Functional Testing Tools, Code Coverage Tools, Test Management Tools, and Performance Testing Tools.

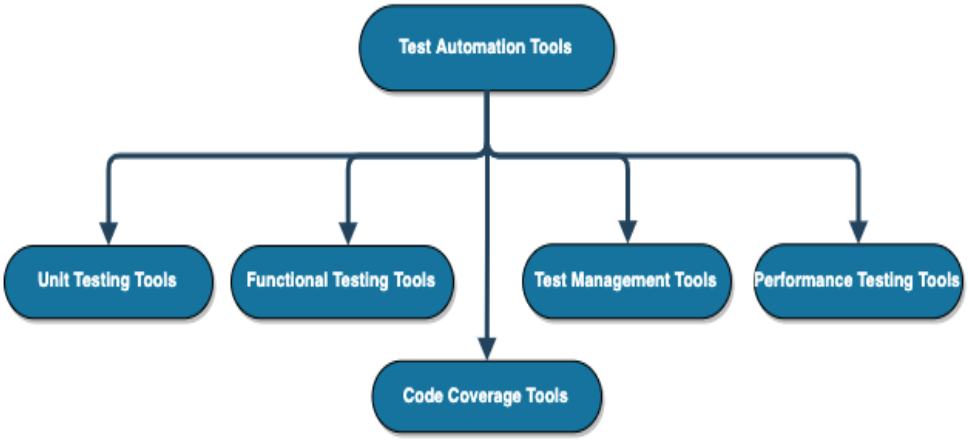

Figure 1: Categories of Test Automation Tools

1) Unit Testing Tools

Unit testing involves testing the most basic units of code, the Unit Testing tools are used to aid unit testing process. Practically, unit testing tools are the most used tools to automate tests [12] and are easily integrable as a framework within a development environment such as NetBeans. Developers can write test cases in an executable language and the test cases can be executed automatically. Unit testing tools are used to tests the correct working of a particular unit (or method) as well as to check code structure and ensure correct observance of programming practice [15] and also ensure correct working of individual units [14]. Some of the unit testing frameworks are: JUnit, NUnit, JMockit, PHPUnit, and soon.

2) Functional Testing Tools

Functional testing is testing performed to ensure that software is functioning as per users' requirements. Functional testing bases its test cases on the specifications of the software component under test. Functional testing tools are tools that are used in functional testing process. Functional testing tools test functions by feeding them input and examining the obtained output in comparison with the specified test oracle (expected output) in the given test case. Functional testing tools evaluate the compliance of a program with specified requirements [16]. Some of the functional testing tools are: Selenium, HP QuickTest Professional, TestComplete, Ranorex, Watir, Tricentis Tosca Testsuite, Test Studio, and soon.

3) Code Coverage Tools

Code Coverage Tools are testing tools that are used to determine parts of software code covered by automated tests. It measures the number of lines, statements, or blocks of code tested using automated test suites. Code coverage testing is an essential metric to understand the quality of Quality Assurance (QA) efforts made [17]. As test cases are developed, code coverage tools highlight aspects of the code which may not be adequately tested and which require additional testing [18], it shows how much of software code is not covered by automated tests and is therefore vulnerable to defects. Code coverage tools typically measure code coverage in percentage values – the closer to 100%, the lower the chance of having undetected software bugs [19]. Some of Code Coverage Tools are: Cobertura, CodeCover, EMMA, PITest, Atlassian Clover, and soon [17].

4) Test Management Tools

Test management tools are used to automate testing activities (such as test cases creation, test plans, test strategy, test results, test reports and soon), and help teams manage projects easily by providing a searchable and maintainable placeholder for test activates. Various Test management tools have a diverse set of features with distinct ways of managing testing. But they commonly offer the possibility of streamlining the testing procedure and permit snappy access to data analysis, and simple correspondence over different venture groups [20]. The following are some of the test management tools: Test Manager, Test Link, TETware, Test Environment Toolkit (TET), QA Complete, and soon.

5) Performance Testing Tools

Performance testing tools are used to aid performance testing processes. Performance Testing is testing perform to determine how the software will perform in terms of responsiveness and stability under various conditions and workload. It can also serve to investigate, measure, validate or verify other quality attributes of software, such as scalability, reliability, and resource usage [21]. Performance testing tools are used to evaluate the performance of software or component in resource usage, throughput and stimulus-response time with specified performance requirement. According to Khan [22] Load and Stress testing are the two performance testing; [23] added Endurance and Spike testing, while [21] added: Soak, Breakpoint, Configuration, Isolation, and Internet testing among the other performance testing types. Performance testing can be done using a wide variety of tools such as: JMeter, Rational Performance Tester, HP LoadRunner, Silk Performer and soon.

III. TEST AUTOMATION FRAMEWORK

A test automation framework is a defined, extensible support structure within which the test automation suite is developed and implemented [24]. Automation Framework comprises of a combination of tools and practices that are designed to help in executing software testing more efficiently [25]. It includes the physical structures used for test creation and implementation as well as the logical interaction between components such as coding standards, test-data handling methods, object repositories, processes for storing test results, or information on how to access external resources. A framework "facilitates a standard way for modifying, adding, and deleting the [test] scripts and functions," and provides "scalability and reliability with less effort" [26].

A. Importance of Test Automation Framework

While testers can still script or record tests using an automation tool, however, integrating it with an organized framework typically provides additional benefits. A well-defined test automation framework will help the testing team in; achieving higher reusability of test components, developing scripts that are easily maintainable and, obtaining high-quality test automation scripts. Utilizing a framework for automated testing will also increase test speed and efficiency, improve test accuracy, and reduce test maintenance costs as well as lower risks [25]. Test automation frameworks provide support foundation to a variety of automated software tests including: Unit testing, Functional testing, Performance testing, and soon [27].

B. Types of Test Automation Framework

There are various types of test automation framework with each having its architecture and may differ from each other based on their support to different automation key factors like reusability, ease of maintenance and soon [24]. It's important to choose the right framework for automating software testing. Some of the most common test automation frameworks are explained below.

1) Linear Automation Framework

This is a first-generation and the simplest testing framework that is used to test a web application's user interface (UI). Here, the tester records each step such as navigation, user input, or checkpoints, and then plays the script back to conduct the test in a sequential order without the need to write code to create functions. It's also known as a record-and-playback framework [25]

2) Modular Based Framework

This is a type of test automation framework in which software under test is divided into separate functions, modules, or sections, each of which will be tested in isolation. A test script is created for each part and then combined to build larger tests in a hierarchical approach. These larger sets of tests will begin to represent various test cases. The key to achieving modularity is by decomposing the functionality and recombining the modules [26]. The modular automation framework is also known as functional decomposition framework [27].

3) Library Architecture Framework

This framework is based on the modular framework but has some additional benefits. It identified similar tasks or functions within the software that need to be tested and grouped them by function instead of dividing the software under test into various independent modules or units to be tested in isolation each with its test scripts. Thus, the software is ultimately broken down by common functions or objectives. These functions are kept in a library which can be called upon by the test scripts whenever needed [25]. Compared to the Modular Based Framework, this framework has a higher degree of reusability because there is a library of common functions that can be used by multiple test scripts.

4) Data-Driven Framework

The Data-driven test framework is a prominent practice in software testing in which test data are separated from script logic and stored externally to an external data source, such as Text Files, Excel Spreadsheets, CSV files, SQL Tables, or ODBC repositories. Test scripts are connected to the external data source and told to read and populate the necessary data when needed [25]. Unlike in Linear Automation, Modular-based, and Library Architecture frameworks, this framework separates test data from the test script, thus allowing testers to test the same function or feature of software multiple times with different sets of test data without the need to modify the test script.

5) Keyword-Driven Framework

This framework follows a similar approach to data-driven framework, in such a way that the test data and script logic are also separated, but this approach takes it a step further. With this approach, keywords are also stored along with their associated objects in an external data source, making them independent from the automation tool being used to execute the tests. Keywords are part of a script representing various actions being performed to test the software. These keywords can be labeled as simply as 'click,' or 'login,' or with complex labels like 'clicklink,' or 'verifylink', and the objects can be a 'Submit Button,' or 'Login Username.' For this approach to work properly, a shared object repository is needed to map objects to their associated actions [25].

6) Hybrid Testing Framework

The hybrid framework is a combination of two or more frameworks set up to get the best practices of different frameworks suitable for automation needs. It leverages the advantages of some frameworks and avoid weaknesses of others. Every software is different, and so should the processes used to test them. As more teams move to an agile model, setting up a flexible framework for automated testing is crucial. A hybrid framework can be more easily adapted to get the best test results [25].

The process of framework evaluating and selecting requires detailed planning and effort. When choosing a framework, it is crucial to consider the right test automation tools to implement the framework with and ensure that it can easily accommodate various automation tools and changes in the software under test thus, achieving smooth tools integration and ultimately helping in achieving successful testing.

IV. AUTOMATION TOOLS

An automation tool is a software itself with the help of which the actual software in focus can be tested, in other words, the automation tool help and serves as a means in doing software testing [28]. The rapid and unparalleled change in technology affects how organizations develop, validate, deliver, and operate software products. Meeting the demands of today's software quality standard requires testing the software, and the success of the testing project is largely determined by testing technique and automation tool used [24]. There are various testing automation tools each having its strengths and weaknesses and serving for a different purpose. A detailed analysis of those various tools should be performed before selecting any tool. Budget, application type, testing requirements, skills required to use the tool are among the factors need to be considered [29]. The investigation process requires a lot of effort, time, and planning. However, the time and effort spent during tool evaluation can go a long way in ensuring a successful testing project [30]. There follows an explanation of some of the most commonly used automation tools along with their advantages and disadvantages.

1) JUnit

JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development, and it allows the developer to write an oracle for each test case, and to automatically execute test sets. JUnit in particular permits to automatically regression test several test sets. If a formal specification is available, it can be translated into assertions which can be checked at runtime, and thus serve as a test oracle [31]. The JUnit framework is one of a family of unit testing frameworks that are collectively known as xUnit that originated with SUnit [32].

Table 2: Advantages and Disadvantages of JUnit

Advantages	Disadvantages
Facilitates Changes and Simplifies Integration	Requires programming skills and consume time to write and check.
Bugs can be found and resolved early without affecting other pieces of the code.	It can only run tests on one JVM (Java virtual machine), as such developers are unable to test applications that require multiple JVMs
Makes the coding process more Agile	Unit tests tools only automate the testing of functionality at the unit level

2) Selenium

Selenium is a framework for testing web applications that is compatible with various browsers and platforms like Windows, Mac, and Linux. Selenium helps the testers to write tests in various programming languages like Java, PHP, C#, Python, Groovy, Ruby, and Perl. It offers record and playback features for doing tests without the need to learn test scripting language [33]. Selenium is perhaps the most popular automation framework that consists of many tools and plugins for Web application testing. Selenium is known for its powerful capability in performance testing and is a popular choice in open-source test automation space, partly due to its large and active development and user community [34].

Table 3: Advantages and Disadvantages of Selenium

Advantages	Disadvantages
Open source, no licensing and maintenance fees.	New teams need to invest time upfront for setup and integration.
Large and active development and user community to keep pace with software technologies.	Slow support from the community.
Open for integration with other tools and frameworks to enhance its capability	Required good programming skills and experience to set up and integrate Selenium with other tools and frameworks.

3) Unified Functional Testing (UFT)

UTF, formerly QuickTest Professional (QTP), is a test automation tool for functional and regression testing, it's probably the most popular commercial tool for functional test automation [33]. UFT offers a comprehensive set of features that can cover most of functional automated testing needs on desktop, mobile and Web platforms. Visual Basic Scripting Edition scripting language is used by this tool to register test processes, operate various objects and control in testing the applications [27].

Table 4: Advantages and Disadvantages of Unified Functional Testing (UTF)

Advantages	Disadvantages
Mature, comprehensive automated testing features integrated into a single system.	Costly solution: license and maintenance fees are considerably high.
Requiring only basic programming skills to get started with test creation and execution.	Supporting only VBScript.
Dedicated user support plus an established large user community.	Possible high costs for upgrades and additional modules.

4) Katalon Studio

Katalon Studio is an automated testing platform that offers a comprehensive set of features to implement full automated testing solutions for Web, API, and Desktop and Mobile applications. Built on top of the open-source Selenium and Appium frameworks, Katalon Studio allows teams to get started with test automation quickly by reducing the effort and expertise required for learning and integrating these frameworks for automated testing needs [35].

Table 5: Advantages and Disadvantages of Katalon Studio

Advantages	Disadvantages
No licensing and maintenance fees required	Poor Community support
Integrating necessary frameworks and features for quick test case creation and execution.	Support limited features
Built on top of the Selenium framework but eliminating the need for advanced programming skills required for Selenium.	Lack of choices for scripting languages: only Java/Groovy is supported.

5) TestComplete

TestComplete is also a commercial integrated platform for desktop, mobile and Web application testing. It enables testers to build a robust testing framework that utilizes the broad spectrum of available software testing methodologies [23]. Like UFT, TestComplete offers some key test automation features such as keyword-driven and data-driven testing, cross-browser testing, API testing, and CI integrations. This tool supports many languages including JavaScript, Python, VBScript, JScript, DelphiScript, C++Script, and C#Script for writing test scripts [36].

Table 6: Advantages and Disadvantages of TestComplete

Advantages	Disadvantages
Many scripting languages to choose from.	Additional fees for extra modules and add-ons.
Only basic programming skills needed.	
Mature, comprehensive automated testing features integrated into a single system.	Like UFT, considerable licensing and maintenance fees needed for TestComplete.

A. Comparison of Automation Tools

There are various automation tools available in the market. Identification of the right automation tool is critical to ensure the success of the testing project. The table below presents a comparison of some of the popular testing tools discussed.

Table 7: Automation Tools Comparison

Features	**Katalon Studio**	**Selenium**	**UFT**	**TestComplete**
Test development platform	Cross-platform	Cross-platform	Windows	Windows
Application under test	Web, Mobile apps, API/Web services	Web apps	Windows desktop, Web, Mobile apps, API/Web services	Windows desktop, Web, Mobile apps, API/Web services
Scripting languages	Java/Groovy	Java, C#, Perl, Python, JavaScript, Ruby, PHP	VBScript	JavaScript, Python, VBScript, JScript, Delphi, C++ and C#
Programming skills	Not required. Recommended for advanced test scripts	Advanced skills needed to integrate various tools	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts
Learning curves	Medium	High	Medium	Medium
Ease of installation and use	Easy to set up and run	Require installing and integrating various tools	Easy to set up and run	Easy to set up and run
Script creation time	Quick	Slow	Quick	Quick
Object storage and maintenance	Built-in object repository, XPath, object re-identification	XPath, UI Maps	Built-in object repository, smart object detection, and correction	Built-in object repository, detecting common objects
Image-based testing	Built-in support	Require installing additional libraries	Built-in support, image-based object recognition	Built-in support
DevOps/ALM integrations	Many	No (require additional libraries)	Many	Many
Continuous integrations	Popular CI tools (e.g. Jenkins, Teamcity)	Various CI tools (e.g. Jenkins, Cruise Control)	Various CI tools (e.g. Jenkins, HP Quality Center)	Various CI tools (e.g. Jenkins, HP Quality Center)
Test Analytics	Katalon Analytics	No	No	No
Product support	Community, Business support service, Dedicated staff	Open source community	Dedicated staff, Community	Dedicated staff, Community
License type	Freeware	Open source (Apache 2.0)	Proprietary	Proprietary
Cost	Free	Free	License and maintenance fees	License and maintenance fees

B. Selection of Automation Tool

There is no good or bad automation tool; hence, the choice of automation tool largely depends on the nature of the software to be tested. For example, Selenium may be the most popular automation tool, but if the software in focus is desktop-based, this tool has no use here. Thus, proper understanding of the testing requirements for the software in focus comes first, then searching for the appropriate tool or often combination of tools that match most of those requirements. Detailed analysis based on the requirements and other certain criteria must be conducted before selecting the tool [24]. The effort put in the tool evaluation process can determine the successful execution of the testing project. The selection of the tool depends on various factors such as:

- The software and its technology stack which is to be tested
- Detailed testing requirements
- Skill sets available in the organization
- License cost of the tool

V. CONCLUSION

Test automation has become an essential part of a successful software testing. The latest World Quality Report 2018–2019 highlights that test automation is the biggest bottleneck to deliver "Quality at Speed," as it saves time, reduces cost, improves efficiency, and increases accuracy [37]. Thus, effective and successful test automation cannot be achieved without the right automation tools and framework, this report presents a detailed explanation about various test automation tools and frameworks as well as provides insights into some of the important factors to consider when selecting automation tool and framework.

REFERENCE

[1] G. J. Myers, C. Sandler, and T. Badgett, The Art of Software Testing 3rd Edition, Third Edit. Canada.: John Wiley & Sons, Inc., 2012.
[2] D. R. Graham, 'TESTING, VERIFICATION AND VALIDATION', Int. J., vol. XVI, pp. 1069–1101, 1979.
[3] S. Rogerson, 'The Chinook Helicopter Disaster', 2002. [Online]. Available: https://www5.in.tum.de/~huckle/chinook_software.pdf.
[4] T. Gang, 'A Collection of Well-Known Software Failures'. [Online]. Available: http://www.cse.psu.edu/~gxt29/bug/softwarebug.html. [Accessed: 09-Sep-2019].
[5] C. Jones, 'Software Quality in 2012: a Survey of the State of the Art', 2012.
[6] NIST Report, 'The Economic Impacts of Inadequate Infrastructure for Software Testing', 2002.
[7] L. Luo, 'A Report on Software Testing Techniques', Pittsburgh, USA.
[8] A. Dennis, B. H. Wixom, and D. Tegarden, Systems Analysis and Design with OOP Approach with UML 2.0, 4th Editio. USA: John Wiley & Sons, Inc., 2009.
[9] A. Dennis, B. H. Wixom, and R. M. Roth, Systems Analysis and Design 5th Edition, 5th Editio. USA: John Wiley & Sons, Inc., 2012.
[10] C. Padmini, 'Beginners Guide To Software Testing', pp. 1–41, 2013.
[11] 'Automated Software Testing - International Software Test Institute'. [Online]. Available: https://www.test-institute.org/Automated_Software_Testing.php. [Accessed: 18-Nov-2019].
[12] M. Polo, P. Reales, M. Piattini, and C. Ebert, 'Test automation', IEEE Softw., vol. 30, no. 1, pp. 84–89, 2013.
[13] M. A. Umar, 'Comprehensive study of software testing : Categories , levels , techniques , and types', International Journal of Advance Research, Ideas and Innovations in Technology, vol. 5, no. 6, pp. 32–40, 2019.
[14] D. Huizinga and A. Kolawa, Automated defect prevention. Hoboken, N.J.: Wiley-Interscience, 2007.
[15] 'difference between unit functional acceptance and integration testing - StackOverflow'. [Online]. Available: https://stackoverflow.com/questions/4904096/whats-the-difference-between-unit-functional-acceptance-and-integration-test.
[16] 'Functional Testing Wikipedia'. [Online]. Available: https://en.wikipedia.org/wiki/Functional_testing.
[17] Stackify, 'Code Coverage Tools: 25 Tools for Testing in C, C++, Java'. [Online]. Available: https://stackify.com/code-coverage-tools/. [Accessed: 29-Sep-2019].
[18] Atlassian, 'About Code Coverage - Atlassian Documentation', 2017. [Online]. Available: https://confluence.atlassian.com/clover/about-code-coverage-71599496.html. [Accessed: 29-Sep-2019].
[19] SeaLights.io, 'Code Coverage vs. Test Coverage: Pros and Cons | SeaLights'. [Online]. Available: https://www.sealights.io/test-metrics/code-coverage-vs-test-coverage-pros-and-cons/. [Accessed: 29-Sep-2019].
[20] K. Sneha and G. M. Malle, 'Research on software testing techniques and software automation testing tools', 2017 Int. Conf. Energy, Commun. Data Anal. Soft Comput. ICECDS 2017, pp. 77–81, 2017.
[21] 'Performance Testing Wikipedia'. [Online]. Available: https://en.wikipedia.org/wiki/Software_performance_testing.
[22] E. Khan, 'Different Forms of Software Testing Techniques for Finding Errors', Int. J. Comput. Sci. Issues, vol. 7, no. 3, pp. 11–16, 2010.
[23] Aebersold Kirsten, 'Software Testing Methodologies'. [Online]. Available: https://smartbear.com/learn/automated-testing/software-testing-methodologies/. [Accessed: 10-May-2019].
[24] H. Bajaj, 'CHOOSING THE RIGHT AUTOMATION TOOL AND FRAMEWORK', Bengaluru, India.
[25] Aebersold Kirsten, 'Test Automation Frameworks'. [Online]. Available: https://smartbear.com/learn/automated-testing/test-automation-frameworks/. [Accessed: 26-Aug-2019].
[26] A. Bhargava, Designing and implementing test automation frameworks with QTP : learn how to design and implement a test automation framework block by block. Packt Publishing, 2013.
[27] 'A Guide to Automation Frameworks |Smartsheet'. [Online]. Available: https://www.smartsheet.com/test-automation-frameworks-software. [Accessed: 30-Aug-2019].
[28] W. E. Perry, Effective Methods for Software Testing: Includes Complete Guidelines, Checklists, and Templates, Third Edit. 2007.
[29] S. Nidhra and J. Dondeti, 'Black Box and White Box Testing Techniques', Int. J. Embed. Syst. Appl., vol. 2, no. 2, pp. 29–50, 2012.
[30] C. Abraham, 'Test Automation Tool Evaluation', Tamil Nadu, India.
[31] C. Oriat, 'Jartege: A tool for random generation of unit tests for Java classes', Springer-Verlag Berlin Heidelb., pp. 242–256, 2005.
[32] 'JUnit - Wikipedia'. [Online]. Available: https://en.wikipedia.org/wiki/JUnit. [Accessed: 19-Nov-2019].
[33] Altexsoft.com, 'Comparing Automated Testing Tools: Selenium, TestComplete, Ranorex, and more | AltexSoft', 2018. [Online]. Available: https://www.altexsoft.com/blog/engineering/comparing-automated-testing-tools-selenium-testcomplete-ranorex-and-more/. [Accessed: 19-Nov-2019].
[34] 'SeleniumHQ Browser Automation'. [Online]. Available: https://selenium.dev/. [Accessed: 19-Nov-2019].
[35] 'A Comparison of Automated Testing Tools | Katalon Studio'. [Online]. Available: https://www.katalon.com/resources-center/blog/comparison-automated-testing-tools/. [Accessed: 17-May-2019].
[36] Brian, 'Best Automation Testing Tools for 2020 (Top 10 reviews)', 2017. [Online]. Available: https://medium.com/@briananderson2209/best-automation-testing-tools-for-2018-top-10-reviews-8a4a19f664d2. [Accessed: 20-Nov-2019].
[37] worldqualityreport.com, 'World Quality Report 2019', 2018.

AUTHORS PROFILE

Mubarak Albarka Umar received BSc. (Honors) Degree in Software Engineering from the University of East London in 2015, currently pursuing MSc. Degree in Computer Applied Technology at Changchun University of Science and Technology. His research interests include Data Mining, Soft Computing, and Software Engineering (mainly System Analysis and Software Testing).

Zhanfang Chen, Ph.D., Associate Professor of School of Computer Science and Technology at Changchun University of Science and Technology. His research interests include Network Engineering, Software Engineering, and Computer Architecture.