An overview of some tools for automated testing of software applications

Flaviu FUIOR

National Institute for Research and Development in Informatics - ICI Bucharest

flaviu.fuior@ici.ro

Abstract: Software testing is one of the most, if not the most critical phase of the Software Development Life Cycle and this is because it is the phase in which are measured the quality assurance criteria and the Key Performance Indicators already set for releasing software in the production environment. The paper presents an analysis of the most comprehensive seven tools used for automated testing with the purpose to offer an objective approach for choosing an automated testing tool. In the paper, one can find, in a structured manner, a short description of the tools under investigation with advantages and disadvantages followed by a synthetic comparison between them based on eight criteria.

Keywords: Software Development Lifecycle; Software Testing; IT Quality Assurance; Automated Testing Tool.

O imagine de ansamblu a unor instrumente pentru testarea automată a aplicațiilor software

Rezumat: Testarea software-ului este una dintre cele mai importante, dacă nu chiar cea mai critică etapă a ciclului de viață al dezvoltării software-ului și aceasta deoarece este faza în care sunt măsurate criteriile de asigurare a calității și indicatorii cheie de performanță deja stabiliți pentru instalarea software-ului în mediul de producție. Lucrarea prezintă o analiză a celor mai complete șapte instrumente utilizate pentru testarea automată, cu scopul de a oferi o abordare obiectivă pentru alegerea unui instrument automat de testare. În lucrare se poate găsi o scurtă descriere într-o manieră structurată a instrumentelor investigate cu avantaje si dezavantaje, urmată de o comparație sintetică între ele pe baza a opt criterii.

Cuvinte cheie: ciclul de viață al dezvoltării software-ului; testarea software-ului; asigurarea calității în TI; instrument de testare automată.

1. Introduction

Due to digitalization, the use of applications (web, mobile, and desktop) is now present in our daily lives with an increasing frequency. A large number of applications competing with each other in any domain and the desire to come to the market with new features as soon as possible, lead to rethinking and reorganizing the software development life cycle.

During the last two decades, software testing, as part of the software development life cycle, knew an exponential growth in both directions of types of testing and volumes of testing.

One of the most dynamic parts of software testing is automated testing and the paper presents an overview of this domain with the purpose to be useful as a possible guide for choosing a tool for applications automated testing. In the second section are presented the selection criteria, based on which the seven tools described further in the paper were chosen. The third section of the paper briefly describes, in a structured manner, the tools under investigation with their benefits and downsides. In the fourth section, a synthetic comparison between these tools based on eight criteria is presented, as a model for an objective approach in choosing a tool for software applications automated testing. Paper ends with a conclusions section showing that this comparative study can be the basis for developing a model for facilitating the selection of the most appropriate tools upon the user's requests.

2. Automated testing tools for software applications

The software release cycle is shorter than ever. Thus, quality is often sacrificed for speed. Testing helps achieve numerous benefits, including the confidence in the delivered code quality and increased application reliability. Test Automation accelerates regression times and reduces testing costs with the expert testing teams.

https://doi.org/10.33436/v29i3y201908

The most used automated testing types are Automated Unit Tests, Automated Functional and Non-Functional Tests, Automated Regression Tests, Automated Deployments.

The automated testing will provide the following benefits: faster release cycles, better code quality, better test coverage, better reliability.

Automated testing must be realized using proper tools that provide advanced capabilities for optimization, execution, and reporting. Of course, in the market, many tools provide automated testing, and this study is focused on seven of them that were chosen for the following reasons:

- these are tools that provide at least functional testing capabilities for applications using automated testing.
- the selection did not include the financial aspect (reported software revenue) like in Gartner (Gartner, 2017, 2018), and there are included free applications.
- the selection was based on how well known and used are the tools in the test automation market (Katalon, 2019).
- are presented together new and mature applications based on the criteria of being comprehensive automation tools and not selecting niches tools like Load Testing tools (e.g., Apache JMeter) or Mobile Test Automation tools (e.g., Appium - a tool for automated testing for mobile applications that allows testing for the newest mobile devices without any additional installations; in this way, it allows to speed up mobile testing by using objects detection and storage (Altexsoft, 2019)).

3. Tools description with advantages and disadvantages

The tools presented below are operating on different systems/platforms like Windows, Mac OS X, Linux; are used for testing different applications such as Web (UI&API), Mobile, Desktop; and are using a large variety of scripting languages like Java, Python, C#, JavaScript, PHP, Ruby, Perl, Groovy, VBScript, C++, JScript, VB, Ruby on Rails, Delphi, Angular, VB .Net, IronPython. These tools can be used by advanced testers and in some cases by novice users (as programming skills) and can have a mild, moderate, or steep learning curve.

We focused on seven tools for applications testing automation: Selenium, Katalon Studio, Unified Functional Testing, TestComplete, IBM Rational Functional Tester, Ranorex Studio, and Tricentis Tosca. Characteristics, advantages, and disadvantages of using these tools are presented below.

Selenium

Selenium is a collection of test automation instruments that has become the most well-known and used product in the quality assurance market in the last decade (Idatalabs, 2019). Presently, Selenium is used as the main automated-testing tool in many small and big companies, because it supports all main operating systems, has a long list of programming languages for writing testing scripts, and supports many browsers. Selenium came with an innovative concept that allowed testers to speak with an under-testing application using programming language they want (Altexsoft, 2019).

Selenium is a collection of instruments that consists of:

- *Selenium Integrated Development Environment* (IDE) is a Firefox add-on, which provides an easy-to-use interface for developing and running automated tests without the need to learn a test scripting language (Selenium Project - Selenium Documentation, 2010);
- *Selenium Remote Control* (RC) allows more than simple browser actions and linear execution; it makes use of the full power of programming languages C#, Java, PHP, Python, Ruby, PERL to create complex tests;

- *Selenium WebDriver* (also known as Selenium 2) is used to send commands directly to the browser and to retrieve results;
- *Selenium Grid* is used to accelerate the testing process by running parallel tests across different machines and browsers at the same time in order to minimize execution time (Chandrasekhar, 2018).

Advantages of using Selenium

- it has flexibility by offering many choices in areas like supported languages, platforms, and browsers;
- Selenium is integrated with Continuous Delivery workflow through easily integrates with many development platforms such as Jenkins, QMetry, SauceLabs;
- Selenium has a giant library of plugins that permit it to extend beyond its normal functionality;
- being the most well-known and used tool in the IT quality assurance domain for test automation, Selenium has a large community behind.

Disadvantages of using Selenium

- to be in the current trends in automated testing domain, more and more testing tools provide codeless testing, and this is possible by offering a user interface on top of the code layer and the possibility to switch between these two modes. Selenium is not providing codeless testing functionality;
- no reporting functionalities and missing of automatically generated reports (test outcomes in the structure of charts, tables, and with attaching screenshots) is one of the most significant challenges of Selenium, and for its resolution Selenium has to count on external solutions;
- Selenium is not providing image comparison functionality between a pattern image and the one displayed during test running. Other applications have this functionality built-in, but Selenium needs to rely on external software like Sikuli.

Katalon Studio

Katalon Studio is a quite new but powerful and comprehensive tool for automated testing of the applications. It has a wide range of features that allow automated testing for Web, API, and Mobile applications. This solution is built over the Selenium and Appium frameworks, and the main benefit is diminishing effort and knowledge needed to learn and integrate these frameworks for automated testing purpose.

Katalon Studio does not require additional integrations to test APIs because it has a dedicated module in the application that permits users to run end-to-end API testing. With Katalon Studio can be tested all types of SOAP/1.1, SOAP/1.2 and REST requests and multiple data sources (XML, XLS, databases with dynamic mapping to be capable of maximizing test coverage). This API testing module allows users to import tests from different API testing and editing applications such as Postman and WSDL (Altexsoft, 2019).

This tool has an entire web testing solution with built-in Continuous Delivery / Continuous Integration. It allows all kind of testing for Web and Mobile (Android, and iOS) applications. For mobile applications, Katalon Studio has a natural integration with Appium.

Advantages of using Katalon Studio

- it can be integrated with a large number of applications, used in Continuous Delivery/ Continuous Integration and DevOps, like JIRA, Git, qTest Manager, Slack, Jenkins, TeamCity, CircleCI, Travis CI, SauceLabs, Selenium Grid, and BrowserStack;
- it has a simple and friendly GUI that provides rich graphics, tree views, and menus; thus,

the user, even a non-technical tester, can manage test items quite easily;

- it supports several types of testing: keywords-driven, data-driven, API-testing and cross-browser;
- being a free tool is a considerable advantage, but in order to benefit of support, a pay is compulsory.

Disadvantages of using Katalon Studio

- it supports only a programming language, Groovy, that belongs to Java family; so, any user of Java can use Katalon, but a larger number of programming languages would be an advantage;
- it has a closed source code, resulting a not so large community supported by several developers, but it has some open components. An open-source plugin platform is planned.

Unified Functional Testing (UFT)

UFT is the new name for rebranded QuickTest Professional (QTP) and the tool owner, MicroFocus, is a company that was bought by Hewlett Packard in 2006. In UFT, testing is done using both keywords and manual scripting for functional test and regression test automation (HP Unified Functional Testing, 2016).

UFT has huge popularity, especially in big companies, despite being not a very cheap application. In terms of test automation, UFT has a wide range of capabilities, which includes API, web services, and GUI testing for mobile, web, and desktop applications (Dzone, 2019). UFT has one of the smartest object recognition function in the market. To support continuous integration, UFT can integrate with applications like Jenkins.

Advantages of using UFT

- it allows the tester to use record, play function to record manual testing, and reuse them as automated (Qasymphony, 2019);
- the tool has a very strong partner network which offers support and services to UFT users (Gartner, 2017, 2018);
- it offers a shared repository where the testing team can store artifacts and functions;
- it has a friendly user interface for API testing where API tests, generated from WADL documents, are created, executed, and reported.

Disadvantages of using UFT

- it supports only Windows, for Mac and Linux platforms additional software is required;
- companies that required a lighter instrument for simpler testing needs, may perceive UFT as a too complex tool;
- UFT uses VBScript as scripting language, which is known as one of the simplest programming languages. For advanced users, this is not a considerable disadvantage.

TestComplete

TestComplete is a tool that did not require advanced coding skills and can be used to automate testing for desktop, web, and mobile applications. Using "recording tests" feature, the user has to start recording, perform all the needed actions in the tested application, and after that, TestComplete will automatically convert all the "recorded" actions in a test (Smartbear, 2019).

TestComplete can be equally used for both unit and functional testing; it also offers strong support for regression testing and other types of testing, like data-driven testing or distributed testing.

Advantages of using TestComplete

- TestComplete with an inbuilt editor is simple to use and allows users having any coding knowledge level to add and to delete tests, to modify parameters, to change test order and to run tests. In addition, it allows writing or editing scripts manually if the editor tool is inadequate for the needs;
- it is a powerful and comprehensive tool for testing which supports a large range of languages and frameworks;
- the open COM-based architecture of TestComplete allows extending its functionality using plugins and integrations.

Disadvantages of using TestComplete

- TestComplete cannot be used on Mac OS X and Linux platforms/systems;
- it requires considerable fees for license and maintenance.

IBM Rational Functional Tester (RFT)

RFT is an automated functional testing and regression testing tool based on an object-oriented automated testing software that allows to test applications, and record scripts that can be played back to validate new versions of a test application (Khan, 2015). Test scripts can then be run by RFT to validate application functionality and supports a wide range of applications, such as web-based, .Net, Java, Siebel, Power Builder, Ajax, GEF (Medium, 2019).

RFT tool offers a powerful recorder for recording tester’s actions with multiple customization options available. The tool has smart script maintenance functionalities to help ensure a test creation and execution process it's flexible when changes in the under-testing application are done (IBM Rational Functional Tester, 2008).

Advantages of using RFT

- RFT users have a broad network of skilled resources which provides support;
- it supports a comprehensive DevOps process by test automation and releases automation, enabling continuous delivery with automated testing;
- in RFT, test scripting can be done, by advanced testers, directly in a native Java or Visual Basic .NET editor and debugger. Developers usually find this feature useful because they can access RFT within the IDE used by the development team to build GUI (IBM Rational Functional Tester, 2008).

Disadvantages of using RFT

- RFT's portfolio is comprehensive and maybe could be perceived as too complex, especially in the early stages of test automation or for simpler testing needs;
- customers indicated below-average satisfaction overall with RFT's test automation offering due to the lacking of the support for the latest technologies and modern web UI toolkits.

Ranorex Studio

Ranorex is a comprehensive and robust tool used for desktop (Windows), web, and mobile applications testing, and allows automation as well for inexperienced users and advanced testers. This tool is a proper solution to enforce automated testing in continuous delivery, which demand continuous testing and quick feedback for testing outcomes.

Ranorex Studio tool includes components like Ranorex Recorder, object repository, Ranorex Spy, code editor, debugger and all these elements create a platform, which provides complete flexibility with code-free testing for manual testers as well as a professional API solution, for more complex test automation needs.

Advantages of using Ranorex

- allows testers with less scripting languages knowledge to generate automated tests with Ranorex Recorder;
- it has a built-in image comparison. Ranorex smart object identification mechanism is able to automatically identify any change in the UI by making use of GUI recognition elements (RanoreXPath, Ranorex Spy) (Altexsoft, 2019);
- after each test run, Ranorex provides an automatic report that offers details on test execution and includes visual screenshots for validation. Ranorex reports offer graphics charts that may include current and previous results of the tests (Guru99, 2019).

Disadvantages of using Ranorex

- Ranorex supports just a few programming languages like C#, VB.Net, and IronPython;
- this tool does not support Mac OS X platform;
- only a small community supports Ranorex.

Tricentis Tosca

Tricentis Tosca is one of the few continuous testing platforms which is known to be a comprehensive and powerful toolset to support a wide range of testing activities stretching from test design, to test automation, to test reports and analytics.

This tool offers a very friendly and simple to use GUI and a rich feature set for designing, running, managing, optimizing API tests and supporting best continuous integration (CI) and DevOps practices; it offers multiple facilities like dashboards, analytics, integrations, and distributed executions (Medium, 2019) (Tricentis, 2019).

Advantages of using Tricentis Tosca

- it works perfectly in the Continuous Integration workflow and is able to significantly diminish development time and to run multiple tests a day. This is possible by using scheduling tools to execute test cases directly from them and after that transfer the results back as an XML file;
- the tool has a good set of test reporting and analytics capabilities;
- it has a mild learning curve because this tool was created to be used by persons with non-developer skills and for this reason, it is easy to set up and learn;
- it has built strong relationships with service and technology partners, which led to the certification of a significant number of resources.

Disadvantages of using Tricentis Tosca

- according to customers, the tool’s price could be more attractive;
- it has a limited knowledge base and forums.

4. Analysis Criteria and Comparative Table

Test automation is an essential part of modern software development lifecycles. There are certainly challenges being faced to choose a tool for applications automated testing. The choice is based on several different types of criteria. In this case, three categories of criteria are considered:

- the tools usage area (Target applications, Operating systems/platforms);
- the ease of learning and using the tool (Scripting languages, Required programming skills, Learning curve);

- popularity, maturity, and cost of the tool (Survey: Automation tools used, Available from, Price).

Table 1 shows a synthetic comparison of the previously presented seven tools, based on eight criteria. The order in which the tools are presented takes into account the result from Survey: Automation tools used*.

	Selenium	Katalon Studio	Unified Functional Testing (UFT)	TestComplete	IBM Rational Functional Tester (RFT)	Ranorex Studio	Tricentis Tosca
Target applications	Web	Web (UI&API) Mobile	Web (UI&API), Desktop, Mobile	Web (UI&API), Desktop, Mobile	Web (UI&API) Desktop, Mobile	Web, Desktop, Mobile	Web (UI&API), Desktop, Mobile
Operating systems / platforms	Windows Mac OS X Linux	Windows Mac OS X Linux	Windows	Windows	Windows Linux	Windows	Windows
Scripting languages	Java, Python, C#, JavaScript, PHP, Ruby, Perl	Java/ Groovy	VBScript	C++, C#, Jscript, JavaScript, VB, Ruby on Rails, PHP, Delphi, Angular	Java, VB .Net	C#, VB.Net, IronPython	JavaScript
Required programming skills	Advanced	Minimum / Advanced (for pro scripting)	Minimum / Advanced (for pro scripting)	Minimum / Advanced (for pro scripting)	Advanced	Minimum / Advanced (for pro scripting)	Minimum / Advanced (for pro scripting)
Learning curve	Steep	Mild	Moderate	Mild	Moderate	Moderate	Mild
Survey: Automation tools used *	84%	38%	15%	6%	5%	5%	3%
Available from	2004	2015	1998	1999	2002	2007	2007
Price	Free	Free	$$$	$$$$	$$$$	$$	$$$$

Survey: Automation tools used* - was done on 2291 participants (Katalon users was one of the sources for the survey). KMS Technology and Katalon conducted the survey with ToolsQA as the media sponsor (Katalon, 2019). Multiple choices were allowed when an answer was given. This is the reason why the sum of percentages is greater than 100%.

Analyzing this comparative table, we may briefly conclude the following:

- from the point of view of the types of targeted applications, the following five tools are the most comprehensive: UFT, TestComplete, RFT, Ranorex Studio, Tricentis Tosca.
- Selenium and Katalon Studio can be used on most systems/platforms.
- Selenium and TestComplete have the widest variety of scripting languages that can be used for creating automated testing.
- based on how "user-friendly" are the tools, the tools complexity and the quality of existing user guides, Katalon Studio, TestComplete, and Tricentis Tosca are showing a mild learning curve. These three tools may be used by novice as well as by advanced (as programming skills) testers.

- it should be pointed out that Selenium and Katalon Studio are free automated testing application, but both have important features and can compete with paid applications.
- according to Gartner Magic Quadrant for Software Test Automation 2018 report, UFT is considered to be one of the leaders in automated test domain together with TestComplete and Tricentis Tosca (Gartner, 2017, 2018). Gartner does not take into evaluation any free tools.

5. Conclusions

A decision to use a tool or another will need to take into consideration many factors like the maturity of the testing team and scripting languages known, supported platforms, the learning curve of the application, and available budget. Finally, on a subjective ground, some factors can influence the decision more than others, but this does not mean that not the same logic of decision always applies.

The paper presents an overview of the most powerful and comprehensive tools for the automation in testing software applications currently existing in the quality assurance market, with the purpose to offer possible guidance for people who want to activate in this area of interest. This comparative study can be the basis for developing a model for facilitating the selection of the most appropriate tools according to the user's requests.

In the future, the author intends to extend the list of presented tools and to add new comparison criteria that will help to build a model based on the user's needs.

Acknowledgement

This work was supported by project PN 19 37 04 01"New solutions for complex problems in current ICT research fields based on modelling and optimization", funded by the Romanian Core Program of the Ministry of Research and Innovation (MCI), 2019-2022.

BIBLIOGRAPHY

1. *Altexsoft*. URL: https://www.altexsoft.com/blog/engineering/ (visited on 15/02/2019).
2. Chandrasekhar, K., Taj, S. G. (2018). *Testing Web Application using Selenium Testing Tool with Respect to Test'ng*. In: International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; Vol. 6, Issue IV, April.
3. Dzone. URL:https://dzone.com/articles/top-10-automated-software-testing-tools (visited on 15/02/2019).
4. *Gartner - Magic Quadrant for Software Test Automation* - 2017, 2018. URL: http://clmconsultores.cl/consultores/wp-content/uploads/2018/01/Gartner-MQ-Automatizaci%C3%B3n-Pruebas-2017.pdf ; https://www.gartner.com/en/documents/3894271 ; https://smartbear.com/resources/white-papers/gartner-magic-quadrant-2018/.
5. *Guru99*. URL: https://www.guru99.com/testing-tools.html (visited on 15/02/2019).
6. HP Unified Functional Testing - User Guide - January 2016. URL: https://softwaresupport.softwaregrp.com/doc/KM02051538?fileName=hp_man_UFT12.52_UserGuide_pdf.pdf.
7. *IBM Rational Functional Tester* - IBM Corporation 2008. URL: ftp://public.dhe.ibm.com/software/rational/web/datasheets/Rational_Functional_Tester_Datasheet.pdf.

8. *Idatalabs*. URL: https://idatalabs.com (visited on 15/02/2019).

9. *Katalon*. URL: https://www.katalon.com/resources-center/blog/infographic-challenges-test-automation/ (visited on 15/02/2019).

10. Khan, Rifa Nizam, and Shobhit Gupta (2015). *Comparative Study of Automated Testing Tools: Rational Functional Tester, Quick Test Professional, Silk Test, and Loadrunner*. In: International Journal Of Advanced Technology In Engineering And Science, Vol. No. 03, Special Issue No. 10, February.

11. *Medium*. URL: https://medium.com/@briananderson2209/best-automation-testing-tools-for-2018-top-10-reviews-8a4a19f664d2 (visited on 15/02/2019).

12. *Qasymphony*. URL:https://www.qasymphony.com/blog/100-plus-best-software-testing-tools/ (visited on 15/02/2019).

13. *Selenium Project - Selenium Documentation* – Release 1.0 – 24 February 2010. URL: https://www.academia.edu/37954589/Selenium_Documentation_Release_1.0_Selenium_Project

14. *Smartbear*. URL: https://smartbear.com (visited on 15/02/2019).

15. *Tricentis.* URL: https://www.tricentis.com (visited on 15/02/2019).

Flaviu FUIOR a absolvit Facultatea de Finanțe, Asigurări, Bănci și Burse de Valori din cadrul Academiei de Studii Economice București în anul 2003. În 2005 a terminat un master în management la Școala Națională de Studii Politice și Administrative București. În prezent își desfășoară activitatea în cadrul Departamentului CDI Modelare, Simulare, Optimizare la Institutul Național de Cercetare-Dezvoltare în Informatică - ICI București. Domeniile sale principale de interes sunt: securitate cibernetică, internetul obiectelor (IoT), testare automată, protecția infrastructurilor critice, metodologii pentru managementul proiectelor.

Flaviu FUIOR graduated the Faculty of Finance, Insurance, Banking and Stock Exchanges of the Bucharest Academy of Economic Studies in 2003. In 2005 he completed a Master in Management at the National School of Political and Administrative Studies Bucharest. He is currently working in the Department CDI Modeling, Simulation, Optimization at the National Institute for Research and Development in Informatics - ICI Bucharest. His main areas of interest are cybersecurity, Internet of Things, automated testing, critical infrastructure protection, project management methodologies.