

Proven Cost Effectiveness through Automated Testing

Introduction:

Optimizing cost effectiveness of the product lifecycle and its testing in particular becomes more and more pressing nowadays. Among the different ways of achieving cost effectiveness of the testing processes, test automation is the most productive. It helps us perform several tests within a short period of time as well as minimize the risks related to the incorrect data input. Still, solid results depend on investing in test automation development, while the scope of spent resources differentiates according to the product, platform and checks required.

Customer

Leading provider of cutting-edge scheduling solutions

Industry

Training Management Systems

Challenge

Develop a test framework to automate UI and functional tests execution and make it continuous.

Business Benefits

Develop a test framework to automate UI and functional tests execution and make it continuous.

Challenges Overview

The desired framework had to respond to the following needs:

- Rapid test case creation. The cases should have the possibility of being created as quickly and effortlessly as possible.
- Low cost for test case creation. The main idea behind the approach is the notion of so-called “cheap test cases”.
- Cross-browser support. A test case has to be independent from a browser or a platform on which it is executed.
- Low cost of automation tools. The solution should not require investments in software development licenses.



Procedural Solution Overview

The solution itself bears the name of EWA (ELEKS Web Automation framework). Our approach does not require large investments in software development licenses since it mainly utilizes free-of-charge products such as Selenium and already available and widely spread test case management systems (TMS) like JIRA.

The test case automation is split into three stages:

- **Recording** is a process of recording user's action using Selenium IDE. The Test Automation Engineer records only the steps described in a test case. At this stage there is no need to handle any validation checks. If a test case cannot be executed correctly after recording, all errors should be skipped as well. The main target of this stage is a background creation for future steps. Usually 50% of recorded actions can be reused without any modification while 20% of actions need minor changes. Using the code generation approach, the recorded test case is then converted to C# code file. The file name is the Test ID for the test case. All test cases are stored in VCS.
- **Stabilization** is a stage of the test case code processing and fixing. The Test Automation Engineer repairs defects in object locators, adds delays, etc. To improve performance at this stage, the typical actions are implemented in such functions as `WaitForObjectToAppear`,

`SelectPopupWindows`, `CreateNewBrowserTab`, etc. The main target of this stage is achieving a stable test case execution. No validation steps are implemented here. After stabilization, the test cases are more tamper-resistant during the execution process.

- **Validation** is a process of adding validation checks to verify the expected results of a test. The standard approach of using a Conditional Operator or unit methods (`Assert` etc.) is very time-consuming. ELEKS offers two approaches for validation that help to save time and improve the quality of validation checks:

Batch properties verification – one of the most common testing tasks aimed at verifying that an object's properties contain the expected values. Batch properties' verification allows for verification of several property values simultaneously using a single test instruction. It is inherently the snapshot of a web page or a desktop window which contains all elements and their properties that should be checked.

Validation functions – the pool of functions for common validation operations, such as sorting and filter validation, paging verification, link checking, and time span validation. One usage of the function allows a user to verify each link on a page and check all pages displayed after the applied filter.

ELEKS Web Automation Framework (EWA)

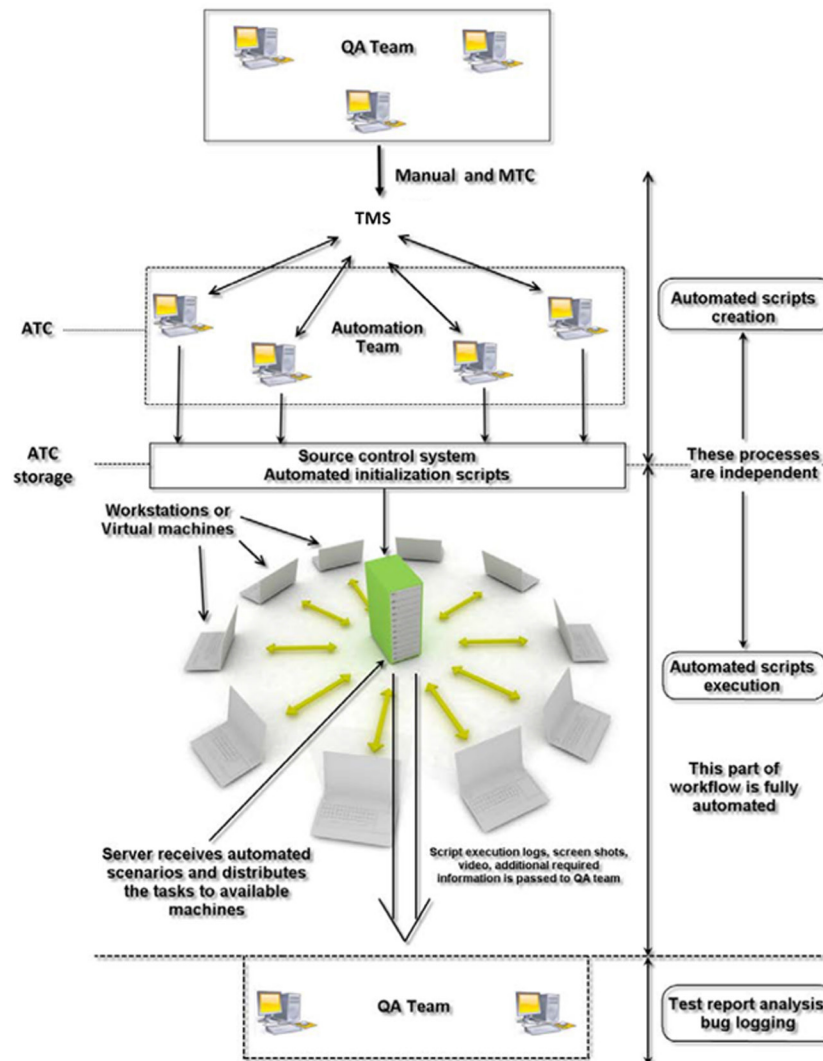
Test case execution is performed in 3 stages:

On a defined schedule EWA connects to TMS using API and extracts all test cases assigned to a certain user specifically created for the automation test cases. Next, EWA groups all test cases and prepares the required virtual machine (VM) images. All VMs are located in ELEKS' private cloud.

- Execution of preconditions
- Test case execution
- Cleanup

EWA generates 3 types of logs:

- Textual log – HTML file contains description of all executed actions with results
- Screenshots – captured bitmap after each step
- Recorded video



Scope of Services and Advantages

- **Manual** – default stage indicates that the current test case should be executed manually by the QA Engineer
- **MTC** – this stage indicates that the current test case should be automated (scripted) by the Test Automation Engineer. Test Automation Engineers automate test cases only at the MTC test stage (not Manual)
- **ATC** – this stage indicates that the current test case is already automated

- Solid, bulletproof EWA framework built upon Selenium
- Product quality assurance across multiple platforms and browsers
- Less efforts dedicated to automated test creation and support
- Maintaining the continuous integration processes
- On-time project delivery

Contact Us:

ELEKS Headquarters

Eleks, Ltd.
7 Naukova St., Building G
Lviv 79060, Ukraine
phone: +380 32 297-1251
fax: +380 32 244-7002

US Office

Eleks, Inc.
701 North Green Valley Pkwy
Suite 200
Henderson, NV 89074
phone: +1 866 588-0113 (toll-free US)
+1 617 600-4059
fax: +1 678 905-9508

email: eleksinfo@eleks.com
www.eleks.com

UK Office

ELEKS Software UK, Ltd.
5 Harbour Exchange
South Quay
London, E14 9GE
phone: +44 203 318-1274