

# Leveraging LeanFT and Selenium WebDriver Together for Legacy ASP.NET Automation Compatibility

Udayan Verma

Denver, USA

[udayanverma7@gmail.com](mailto:udayanverma7@gmail.com)

## Abstract:

The problem of software quality assurance is typical of the businesses, which undergo a nominal digital transformation, as they anticipate receiving new applications with a change implemented and running simultaneously. Even the current models such as the Selenium Web Driver are unsuccessful with the outdated ASP.NET WebForms with dynamic identifiers, rather complex server controls, and page live cycles, causing brittle and hard-to-maintenance tests. This report offers a hybrid automation system with the combined abilities of Selenium (cross-browser) and Micro Focus LeanFT (advanced object recognition) as a way of offering greater power in the automation of both old and modern processes within the system of applications. The framework is developed based on abstraction concepts, modularity, and conditional decision-making to find the appropriate instrument to use in each of the application units. Part of continuous integration and delivery pipelines, it helps in running several processes concurrently, proactive maintenance, and effective test data management.

**Keywords:** Test Automation, Hybrid Framework, Legacy ASP.NET, LeanFT, Selenium WebDriver.

## I. INTRODUCTION

The modernisation of enterprise systems tends to provide a transitional climate where both old and new applications co-exist. The issue of quality assurance at this stage is a major problem, especially in organisations that use ASP.NET WebForms. The standard of the industry, Selenium Web Driver, fails to perform well in this context because of its reliance on locators based on the Domain name. These are usually not resistant to server-side defences and dynamically generated identifiers and create brittle tests which are expensive to maintain.

To overcome these drawbacks, the report will explore an automated system based on a hybrid system that will integrate the cross-browser features of Selenium WebDriver and sophisticated object recognition in Micro Focus LeanFT. This two-tool implementation is applied to the quality assurance ecosystem of Charter Communications to conduct overall testing on the old and the new applications. The discussion below evaluates the justification of this model, the architecture of the system, integration and maintenance of the system, the scale factor, the report system and how it will achieve sustainable automation in the transitional IT environments.

## II. BACKGROUND AND RATIONALE

### A. Challenges of Legacy ASP.NET Automation

ASP.NET WebForms contain some barriers to automation by virtue of their legacy. They dynamically create complex identifiers on the server-side rendering, and the locators are not stable, and the tests are not reliable [1]. Some server-side controls, such as GridViews, TreeViews, and more, usually do not

provide meaningful semantic hooks and therefore require complex and brittle workarounds. Moreover, the dependence on postbacks and elaborate page life cycles is also a problem with ASP.NET, thus making it susceptible to test flakiness. Use of Selenium alone in those conditions results in subsequent maintenance cycles where time goes to waste on locator stabilisation and not working on the business functionality.

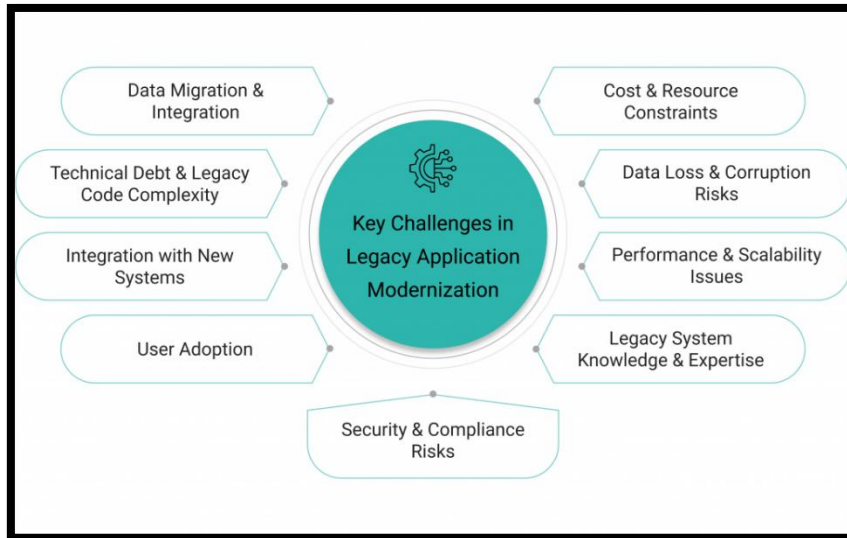


Fig 1: Key Challenges in Legacy Application Modernisation

### ***B. Justification for a Hybrid Model***

The deficiencies of Selenium themselves are the reason why there should be a combined approach. LeanFT provides a powerful object recognition system, and it can query elements by native .NET objects, which will minimise locator brittleness. It can also be integrated with Visual Studio and has extra benefits with Microsoft-based stacks [2]. With the integration of LeanFT with the existing components and Selenium with the recently developed web applications, organisations can experience a healthy and sustainable automation infrastructure. This hybrid approach makes certain that all the tools are used in areas where they are efficient and form a balanced solution, which is stable, maintainable, and provides the coverage of end-to-end tests in mixed technological environments.

## **III. SYSTEM ARCHITECTURE & PIPELINE DESIGN**

### ***A. Unified Architectural Framework***

The design is grounded by the principles of abstraction so that the test scripts would not be dependent on the choice of the tools. Using the Page Object Model (POM), every page or element is described as a class with the interaction logic. Depending on the conditioning, LeanFT or Selenium is called. In the behaviour of legacy ASP.NET sections, LeanFT is used, whereas the current responsive pages are running Selenium WebDriver [6]. This helps keep the scripts clean, readable and dedicated to workflow validation and not tool-specific implementation.

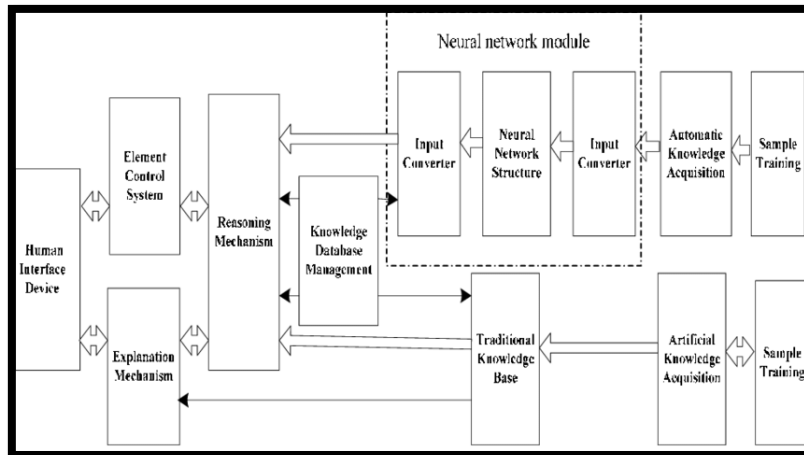


Fig 2: Hybrid Automation System Architecture

### B. Integration into the CI/CD Pipeline

The framework is properly incorporated into a set of CI/CD pipelines that are operated with the help of such tools as Azure DevOps or Jenkins. Here, the automation code is first retrieved, after which an environment is configured, which comprises browsers and LeanFT runtime. Then, tests are run using such runners as NUnit or TestNG, which easily invoke Selenium or LeanFT when needed [3]. At the end of the pipeline, the reports and logs are put into a single output. Through this integration, regression testing is performed with consistency, automation, and high-speed feedback with each build.

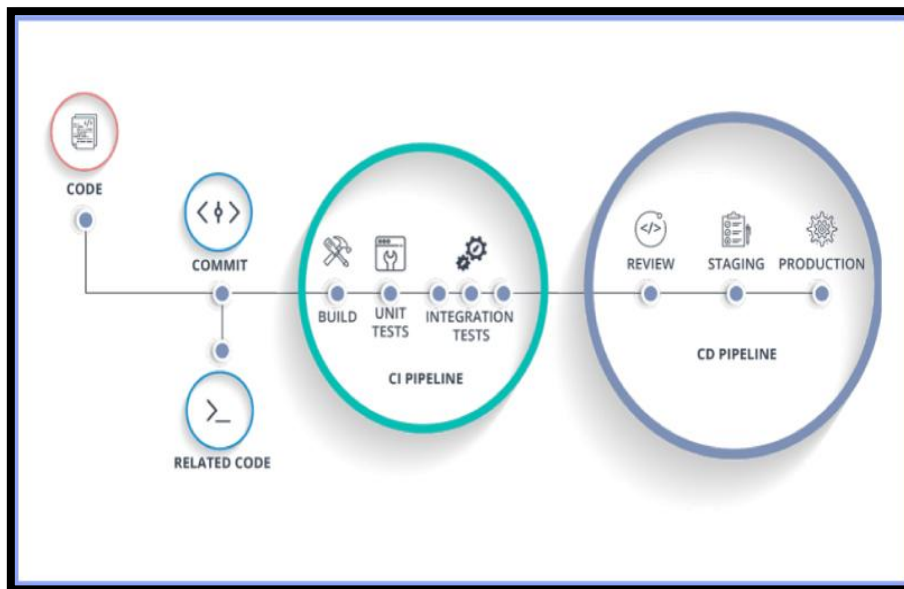


Fig 3: Continuous Integration Pipeline for Test Orchestration

## IV. INTEGRATION AND MAINTENANCE STRATEGY

### A. Targeted Module Development and Integration

This is a successful integration since the application is separated into legacy and modern parts. Specialised LeanFT modules are used to automate automotive WebForms elements, and Selenium modules are used to automate interfaces of the modern world. Each set is defined by a common core library, which offers cohesion [4].

## ***B. Proactive Governance and Maintenance.***

Sustainability is propagated by well-managed maintenance practices. The Selenium elements lean towards not only the stable locators, such as business-facing attributes, but also the use of auto-generated identifiers [5]. The modules of LeanFT exploit multi-property object description to increase resistance to UI changes. Ongoing testing of flaky tests, and consistency and quality receive regulation measures, e.g. peer-review and documentation standards. It is a more orderly method of sustaining long-term maintenance and ensuring the onboarding is successful.

## **V. SCALABILITY AND OPTIMISATION**

### ***A. Achieving Scalability Through Parallel Execution***

The existing development cycles need speedy testing that can be performed by executing them in parallel. Selenium tests leverage Selenium Grid to run tests on a set of browsers and operating systems, whereas LeanFT tests leverage parallel execution test instances on distributed agents. The workloads are smartly divided into the CI/CD pipeline, and the large-scale regression suites can be fully executed at speed. This will save time, and it may also give the needed feedback in time, without omitting the coverage.

### ***B. Continuous Optimisation***

Besides parallelization, optimization is also addressed by improving the performance and efficiency. Wait strategies are fine-tuned to reduce the time spent in execution, and they are stable. Selenium tests begin with the option of starting a headless mode where required and simplifying the creation of the environment. Test data management also removes dependency, and it reduces conflicts within a concurrence environment. Ongoing profiling will reveal the existence of bottlenecks, and it can be streamlined through refactoring of code continuously to become more efficient in memory. All of those will guarantee a high-performance automation system that is able to adjust to the alterations in the business requirements.

## **VI. REPORTING AND MONITORING**

### ***A. Consolidated Reporting and Analytics***

The stakeholders need to view the fruits of automation. Although, though the framework entails the use of two tools, the findings have to must in a single reporting dashboard. Intensive report tools such as Allure or ExtentReports accept standardised results, e.g. JUnit XML. The resulting interactive dashboard gives the high-level execution statistics and provides the opportunity to drill down to specifics of a test, logs, images, and traces of the errors.

### ***B. Monitoring for Continuous Improvement***

The communication aanda system of monitoring may turn into the reports. Trend analysis enables the team to correct the structural problems before they happen since a person can record the repetitive failures or how long it is taking to accomplish that problem is increasing. Application defects too are separated as well as script defects through a step-level report that is triaged effectively and very detailed.

## **VII. CONCLUSION**

Use of Selenium WebDriver may not suffice to have good automation of tests with the traditional ASP.NET apps. They are the inherent challenges of server-side controls, dynamic identifiers and advanced page lifecycles that need a more tailored solution. This is why the hybrid model, which is outlined in this report, is a mixture of Selenium WebDriver and Micro Focus LeanFT and is the solution.

The blend of the greatness of both tools as a highly designed framework will enable organisations to achieve sustainable and scalable automation. Monitoring and Reporting enhance transparency as well as

create continual enhancement. The hybrid model ultimately promotes the integrity of quality assurance throughout the periods of transitional IT environment, hence enabling the businesses to modernise with a high rate of confidence without compromising the integrity of the legacy systems that enhance the popularity of the enterprise.

## REFERENCES:

1. Y. Rachelson, “Using Selenium for Web Automation? Let UFT Pro (LeanFT) help you!,” OpenText DevOps Cloud Blog, Jan. 18, 2018. [Online]. Available: <https://community.opentext.com/devops-cloud/b/devops-blog/posts/using-selenium-for-webautomation-let-uft-pro-leanft-help-you>.
2. OpenText Community, “Easily integrating LeanFT into your Selenium-based projects,” OpenText DevOps Cloud Blog, 2018. [Online]. Available: <https://community.opentext.com/devops-cloud/b/devops-blog/posts/selenium-and-leanft-easily-integrating-leanft-into-your-selenium-based-projects>
3. Wipro Technologies, “Test Automation tool Migration – Commercial to Open Source,” White Paper, 2018. [Online]. Available: <https://www.wipro.com/content/dam/nexus/en/service-lines/applications/latest-thinking/test-automation-tool-migration-commercial-to-open-source.pdf>
4. Euro-Testing Software Solutions, “LeanFT vs Selenium – Our Experience (Infographic),” Euro-Testing Blog, Feb. 26, 2019. [Online]. Available: <https://www.euro-testing.com/software-testing/leanft-vs-selenium/>
5. AlgoShack Technologies, “Test Automation Using LeanFT,” Medium, Jan. 19, 2023. [Online]. Available: <https://medium.com/@algoshack/test-automation-using-leanft-ceb436c2f3c9>
6. M. Leotta, B. García, F. Ricca, and J. Whitehead, “Challenges of End-to-End Testing with Selenium WebDriver and How to Face Them: A Survey,” In Proc. 16th IEEE Int. Conf. Software Testing, Verification and Validation (ICST 2023), pp. 339-350, Apr. 2023, doi: 10.1109/ICST57152.2023.00039.