

# LEVERAGING TEST AUTOMATION FOR CMS MIGRATION

WHITE PAPER



Dileep Balivada  
Associate Manager

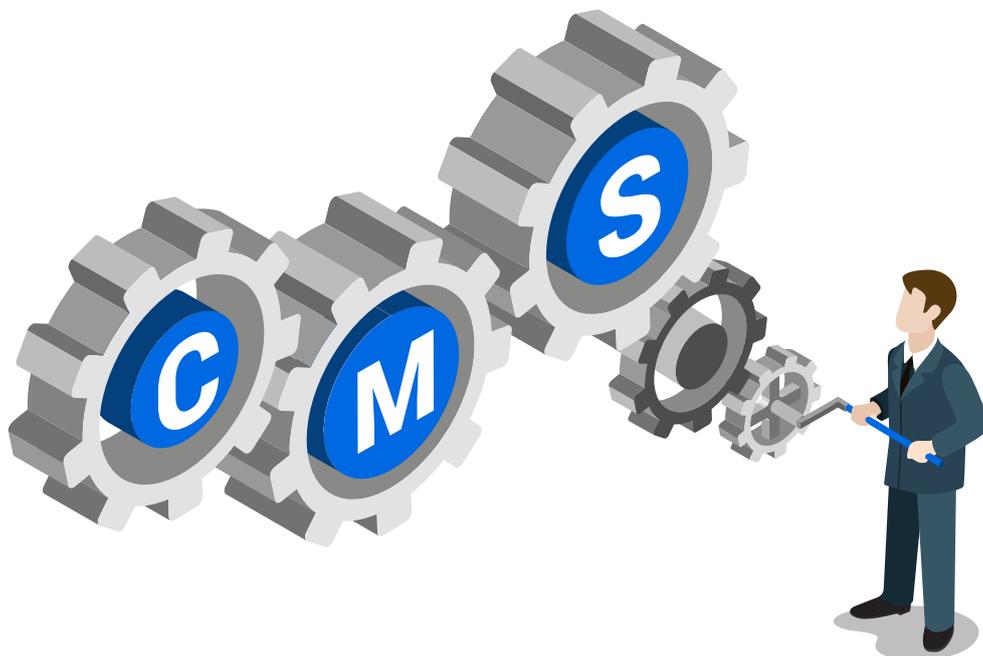
Pradeep Aavula  
Technical Lead

**TA DIGITAL**  
Intelligent Transformation

## TABLE OF CONTENT

---

<b>Introduction</b> .....	2
<b>Approach for functional automation</b> .....	2
Automation feasibility .....	2
Automation framework Identification .....	3
1. Abstraction of test code from test cases .....	3
2. Good Structure .....	3
3. Inherent reusability .....	4
4. Context-related controls .....	4
5. Adaptability .....	4
6. Reporting .....	4
7. Support continuous integration .....	4
8. Test automation in cloud .....	4
<b>Functional testing automation for a key customer</b> .....	5
<b>Other automation utilities built for a key customer</b> .....	6
1. Broken links validation .....	6
2. Vanity URLs checker .....	6
3. Links navigation checker .....	6
4. 301/302 redirects checker .....	6
5. SEO checker .....	6
<b>UI Comparator</b> .....	7
<b>Overall Benefits</b> .....	7



## INTRODUCTION

---

In today's digital world, change is inevitable to meet the scale of new-age business requirements. Most of the IT organizations have discovered the need to upgrade their existing Content Management System to digital CMS, which would enable them to adapt to the demanding needs and manage digital assets/content in a much efficient manner. Besides, to implement new CMS in a quick time, it's essential to conduct extensive testing and ensure that the content is thoroughly validated. A lot of manual effort is required to validate the existing functions, likely new issues such as broken links, redirects, error pages, new HTML tags, to name a few while migration. There are high chances that some of the potential defects may get missed out due to the plethora of browsers, device combinations, and the scale involved. A tester is likely to develop change blindness and fatigue, doing mundane, repetitive tasks. This whitepaper is an attempt to provide insight into a cost-effective test automation solution that caters to overcome the challenges and disadvantages involved in manual testing activities during a CMS migration/upgrade project.

### Leveraging automation as a solution for optimization

In CMS upgrade/migration projects, the first aspect a QA team should focus is on application functionality. Testers need to ensure that the application behavior remains unaffected post-migration/upgrade. However, a manual testing effort is time consuming, tedious, and repetitive as the QA team may have to execute the same set of test scenarios with different test data across different browsers. Hence, there arises a need for automation testing to improve efficiency, reduce manual intervention, and, more importantly, optimize efforts.

## APPROACH FOR FUNCTIONAL AUTOMATION

---

### Automation feasibility

The first logical step towards moving to automation is to identify the possible benefits that can be realized with this exercise. A complete understanding of the application will help in the determination of the test automation scope. Critical business success related test scenarios, most used portions of the application, and stable functionalities are some of the factors that are likely to help to firm up the automation scope. A feasibility study needs to be carried out by a thorough examination of this automation scope to determine the subset that can be automated. The feasibility study will bring out the technical challenges or impediments that can result in keeping out some of the tests identified initially to be excluded from the scope of automation.

Identification of approximate build cycles with regression and smoke testing needs, likely efforts in building and maintaining the tests can help in concluding if it's worth to spend efforts in this endeavor. Usually, a project with a short duration with fewer sprints won't qualify for automation.

## Automation framework identification

An automation test framework and right tool selection are some of the critical success factors for a winning test automation approach. Below are some of the other factors for identifying the right tool for automation:



Test coverage



Cost-effectiveness



Ease of maintenance

Though there are a lot of commercial tools available in the market, most IT organizations have shown an inclination to open-source test automation solutions like Selenium for web applications. However, one should not use Selenium alone with computer languages combination and showcase the test results to client stakeholders, as it is hard to understand the underlying code written in it.

Business users do not possess the technical skills to understand such frameworks and associate the scripts with real-time scenarios. Moreover, there is no standard approach defined on the implementation, as different automation QA testers employ different methods to call functions from different classes. This is where a tool like Cucumber helps. Cucumber is a testing tool that supports the Behavior Driven Development (BDD) framework. It defines application behavior using simple English text, defined by a language called Gherkin. Cucumber allows automation functional validation that is easily read and understood. Cucumber was initially implemented in Ruby and then extended to the Java framework. The biggest advantage of Cucumber is the improved collaboration between technical and non-technical teams. Test cases can be written using real-time examples of the actual requirements to explain the behavior of the system. Hence, most of the IT organizations are now moving to a standard BDD with Cucumber framework to implement test automation, especially in Agile. BDD with Cucumber provides excellent flexibility to come up with test cases (feature files), which form the basis for driving test automation. Below are a few advantages of implementing BDD test automation framework:

### 1. Abstraction of test code from test cases

BDD offers flexibility to modify the test cases without touching the test code. Test cases are the feature files written in English, and the test steps are, in fact, the backbone of the automation. Traditional frameworks generally do not have this isolation even if test methods are loaded with comments.

### 2. Good structure

BDD framework is well structured to segregate test cases, test steps, and the supporting classes. Traditional frameworks can also be structured well, but the onus is on QA Automation Architects/QA Automation Engineers to build a robust and reusable framework.

### **3. Inherent reusability**

Test steps in BDD are developed in such a way that they can be reused by any scenario from any of the feature files. Once enough steps are developed, new tests can reuse the existing steps designed and may not need any additional automation code. Traditional automation frameworks do not have this mechanism and can be implemented by calling functions and classes outside of test classes. However, it is solely dependent on the skills of the QA automation team to build a reusable framework.

### **4. Context-related controls**

BDD provides hooks to insert additional logic around test steps, test scenarios, features, and even the whole suite. Hooks can automatically log steps to Extent reports; instead of QA automation resources, explicitly writing calls for logging in a traditional automation framework.

### **5. Adaptability**

BDD scenarios are easy to update as the product changes. Plain language is easy to edit. Modular design makes changes to the automation code safer. Scenarios can also be filtered by tag names to decide what runs and what doesn't.

### **6. Reporting**

The framework is enhanced to generate extent reports that provide a very detailed status of each test step and report the discrepancies along with the snapshot and the reasons for failure. This form of reporting is even easy for a business user to understand and relate to the failure scenarios.

### **7. Support continuous integration**

Scripts can be easily integrated into CI/CD pipeline so that the tests get triggered automatically for every build deployment and generate an execution report about the quality of the build. Early feedback on the builds, in the form of test execution reports, helps developers to understand the issues (if any) quickly, fix them rapidly, and redeploy a more efficient, reliable, and optimized code in a short time. One can use the tags defined in feature files to control the execution of test cases without touching the code.

### **8. Test automation in cloud**

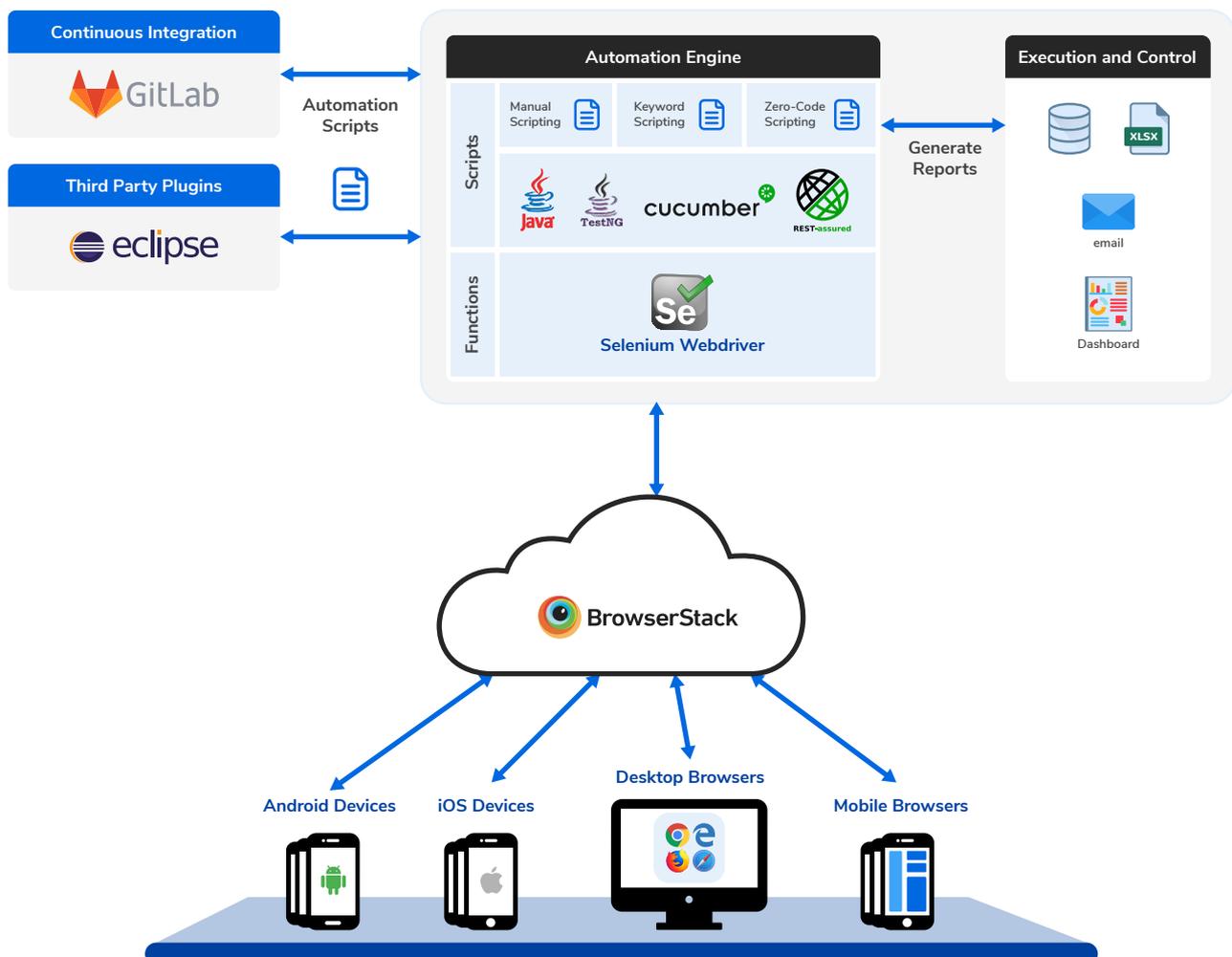
BDD automation scripts support testing in a cloud platform like browser stack/Sauce labs that enables the QA team to test web and mobile applications across on-demand browsers, operating systems, and real mobile devices. This feature will help in more comprehensive test coverage without compromising on security.

## FUNCTIONAL TESTING AUTOMATION FOR A KEY CUSTOMER

One of our customers was facing challenges with the volume of manual testing involved for every release. Moreover, the scope of QA activities was slowly increasing due to new components development and validation of different forms in every release. Hence there was a need to automate smoke tests for build verification and validate the functionality across different browsers and certify the builds on quality in a quick time. In addition to the above need, the customer was expecting a solution that would be easier for business users to understand test scenarios and relate them with business objectives.

To address both technical and business challenges, the TA Digital QA team customized the existing automation framework using Cucumber (for BDD) with Selenium. Test scenarios were written in simple English in Cucumber feature files, and the abstracted test steps were written using Selenium. All automation test scripts were checked in source control (GIT) and integrated with BrowserStack to automate the build and functional validations post-deployment across different browsers both in desktop and mobile.

The below diagram depicts the test automation framework built by the QA team, which has integration with GIT and BrowserStack. The customized version of this framework was used in implementation for the CMS migration test needs for a customer mentioned above.



## OTHER AUTOMATION UTILITIES BUILT FOR A KEY CUSTOMER

---

In general, below are some of the critical checks to be covered by QA team post-CMS migration/upgrade:

- Broken URLs
- Vanity paths
- Font styles
- Spell/grammar check
- Console errors
- Text verification
- Videos verification
- Image verification
- Links navigation
- SEO meta tags
- Structure data
- Accessibility testing
- Analytics
- 301 redirect URLs

The TA Digital QA team built a utility that could target 5 out of 14 checks mentioned above, during the smoke, functional, and regression testing. Following is a quick summary of this utility across the five content testing parameters:

### 1. Broken links validation

Broken links may damage the website's rankings and searchability by preventing search engines from indexing the pages. It also affects user experience by redirecting visitors to error pages. Hence, it's extremely critical to validate all links in web pages and ensure that they are working as intended. However, it requires strenuous efforts to check all links manually in all pages, and the likelihood of not identifying a broken link is high. Hence, an automated approach to crawl all the pages, identifying all the links and check the status, is a more meaningful and safer solution that improves efficiency and reduces manual effort. Broken Link checker built-in Selenium with Java, addresses the above need of automation.

### 2. Vanity URLs checker

Vanity URLs are a type of custom URL that exists to help users remember and find a specific page of your website. Hence, it's important to validate if the Vanity URLs are configured post-migration properly. Checking Vanity URL manually is time-consuming esp. when the volume of pages in scope is huge. Hence an automation script is built using Selenium to automate the validation. All the vanity URLs are captured in test data, and an automation script reads every URL and obtains the corresponding Vanity URL in run time and compares with the expected one.

### 3. Links navigation checker

A lot of manual effort is required to validate the links under different menu items from different pages. Hence, an automation script is built in Selenium that will extract all such links from different pages and see if they are working as intended.

### 4. 301/302 redirects checker

All the 301/302 redirects are also automated using Selenium script that reads the source URL from test data and sees if it is redirected to the target URL.

### 5. SEO checker

SEO checks are very important as it determines the website's rankings and searchability. An automation script in Selenium is built that checks the SEO tags and report all the major discrepancies.

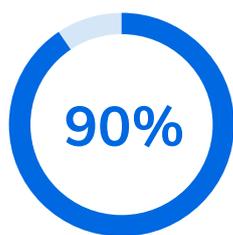
## UI COMPARATOR

Apart from the above checks, for “as-is” migration, it is essential to thoroughly validate user interface, post CMS migration, and ensure that the complete content is getting populated on webpages for different screen sizes. Manual verification is time-consuming, error-prone, and requires additional effort to compare dynamic content in different devices and browsers. Also, manual testing may not be able to identify major differences in an image, if it flickers on and off constantly. Hence, a new automation tool (Visual Regression) is built-in Selenium, which captures a screenshot of both UIs (pre and post) and does a quick pixel to pixel comparison and reports the differences. This utility can quickly make a comparison for multiple pages in one-go and generates a final report with discrepancies. Even the slightest of differences in pixels between pre and post UI screens can be easily spotted and reported. The tool has the capability to perform a similar match based on configured parameters like threshold and delay.

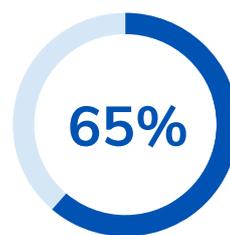
## OVERALL BENEFITS

Following are some of the key benefits realized during functional/regression/smoke testing using the above solutions:

- 90% manual effort savings per regression cycle
- Overall 65% effort and cost savings keeping test automation development effort in consideration
- Increase in test coverage
- Faster feedback and defect detection
- Faster time to market



90% manual effort savings  
per regression cycle



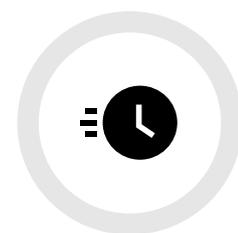
Overall 65% effort  
and cost savings



Increase in  
Test Coverage



Faster feedback and  
defect detection



Faster time to  
market

TA Digital is the only global boutique agency that delivers the “best of both worlds” to clients seeking to achieve organizational success through digital transformation. Unlike smaller, regional agencies that lack the ability to scale or large organizations that succumb to a quantity-over-quality approach, we offer resource diversity while also providing meticulous attention to the details that enable strategic success.

Over the past 20 years, TA Digital has positioned clients to achieve digital maturity by focusing on data, customer-centricity and exponential return on investment; by melding exceptional user experience and data-driven methodologies with artificial intelligence and machine learning, we enable digital transformations that intelligently build upon the strategies we set into motion. We are known as a global leader that assists marketing and technology executives in understanding the digital ecosystem while identifying cultural and operational gaps within their business-ultimately ushering organizations toward a more mature model and profitable digital landscape.

Recognized in 2013, 2014, 2015, and 2019 Inc. 5000 list as one of the most successful technology companies in the United States, TA Digital is pleased also to share high-level strategic partnerships with world class digital experience platform companies like Adobe, SAP and Salesforce and possess global partnerships with industry leaders such as Sitecore, Episerver, Elastic Path, BigCommerce, AWS, Azure and Coveo.



[www.tadigital.com](http://www.tadigital.com) | [sales@tadigital.com](mailto:sales@tadigital.com)

TA Digital™ is a registered trademark of TechAspect Solutions, Inc.

© 2020. No part of this document may be distributed, reproduced or posted without our written permission.