1.  Executive Summary

Automated Testing is a core activity of any agile development methodology. As we move towards continuous deployment, test automation becomes ever more important due to the quick feedback response that it provides to the development team about the health of the application.

In order to get this quick feedback, automated tests need to be executed continuously, should be fast and test results should be consistent and reliable. To achieve these, development and testing should be a coherent activity, and quality should be “baked in” right from the start by ensuring that what is being developed works and that it hasn’t broken existing functionality.

2.  Test Automation Strategy Overview

Quality should be favoured over quantity - It is usually better to release with one feature that is rock solid rather than multiple features that are flaky. As a minimum release criterion, any newly developed feature should not have introduced any regression defects.

Prevention rather than detection - The methodologies are defined to allow quick detection of bugs when they are introduced into the system and feedback to development. A process and a mechanism for obtaining feedback quickly is formulated, which is very important to support continuous delivery.

One good way is to apply low-level tests such as unit test, integration test and API test to ensure the code is working as intended and helps prevent defects escaping in other layers of testing. The second element of improvement is running the regression tests more frequently and aligned with the process of Continuous Integration.

Automation testing should not be seen as an isolated task, but rather as a coherent activity embedded in the software development life cycle.

3.  Automated Test Best Practice

* An automation tool is important, but it is not the solution of everything
* Avoid GUI Automation when there is an alternate present
* You cannot automate everything
* Know the application being tested

4.  Scope of Automated Testing

   4.1 Testing Coverage

* Write the automation test scripts which are to support the regression test and EndToEnd test. The target of Automated Test is to cover 50% manual test cases of each function.

   4.2 Tools

* Automation Test Framework: Cypress
* Editor Tool: Visual Studio Code
* Programming language: Javascript

   4.3 Testing Level

|  |  |  |
| --- | --- | --- |
| **No.** | **Item** | **Detail** |
| 1. | Automated Unit Testing | Out of scope in this document |
| 2. | Automated API Testing | Out of scope in this document |
| 3. | Automated UI Testing | Write test scripts to perform EndToEnd test base on manual test cases |

5.  Test Automation Framework

5.1 Overview

  There are various frameworks to automate an application testing which can be employed as per the need of the application. Every framework has its advantages and disadvantages.

We can choose the Record & Play Back method or descriptive programming after analysing the feature provided by any automation tool, the limitation of the product/application, timeline, and project management methodology:

The following points are considered to define the automation test framework:

* Provide a strategy for design and development of an efficient automation framework in an iterative incremental development model
* Address issues surrounding framework compatibility and productivity with respect to certifications and implementations on multiple technology stacks
* Provide a fit-for-purpose automation solution that is neither too costly nor too scanty in functionality
* Focus on the scalability of the framework, leading to higher efficiency in activities related to enhancing its functional footprint
* Use modularity to assist maintenance and ensure consistent performance in a scenario where the application under test is constantly evolving
* Briefly outline the framework design methodology, customized to needs and maturity of the organization implementing the solution
* Identify key activities that help ensure the project stays on track.

Diagram

Description automatically generated

  5.2 Cypress test framework is selected

We apply the Cypress Testing Framework

**Advantaged:**

* First point and the most important is the price. It is free to use.
* Cypress framework helps in writing the tests which are easy to understand by anyone irrespective of the Technical knowledge they possess
* Cypress framework captures snapshots at the time of test execution
* Cypress framework doesn’t need to add explicit or implicit wait commands in test scripts

**Disadvantaged:**

* One cannot use Cypress to drive two browsers at the same time
* It doesn’t provide support for multi-tabs
* Cypress doesn’t provide support for browsers like Safari and IE at the moment
* Cypress only supports JavaScript/TestScript for creating test cases

5.3 Detail of Automation Test Framework

![Diagram

Description automatically generated]()

5.4 Test Data Management

Automation test data will be managed externally. In case, the test data is changed in future, it should not affect the automated test script.

In this project, the test data is managed in the csv file and the test data will be deleted after the script is run completely.

When the test needs the prerequisite data, the team will try to generate the prerequisite data within automation script wherever possible so that its unique for each automaton run.

The test data will input into the automation test script will cover the following cases:

* No data: Check system response when no data is submitted
* Valid data: Check system response when Valid test data is submitted
* Invalid data: Check system response when Invalid test data is submitted
* Illegal data format: Check system response when test data is in an invalid format
* Boundary Condition Dataset: Test data meeting boundary value conditions
* Equivalence Partition Data Set: Test data qualifying your equivalence partitions.
* Decision Table Data Set: Test data qualifying your decision table testing strategy
* State Transition Test Data Set: Test data meeting your state transition testing strategy
* Use Case Test Data: Test Data in-sync with your use cases.

5.5 Test Execution & Reporting

The team will use command line to execute the ..js files under features folder. The HTML report is automatically generated after the script is run finish. It includes the percentage of passed/failed cases in the result and it has screenshot for the failed case.

6.  Roles and Responsibilities

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Senior automation test engineer | The senior automation test engineer plays a key role in terms of initializing automation work, adopting and implementing framework enhancement, monitoring automation activities as well as mentoring team members. |
| Automation test lead | The automation test lead possesses not only strong technical skills but also ability to manage automation testing teams. This engineer is usually certified advanced software testing qualification and, in some large project, he/she can plays the role of test manager. |

7.  Deliverables

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Automation Test Approach | This document |
| Test case / test suite | Documents that be based to develop test scripts. It might be a copy of manual test suite, or modified for being implemented in terms of automation practices |
| Test framework | Consists of designs, structures and methods for developing test scripts. Test framework is usually delivered with test scenarios, test scripts, test data and their configurations |
| Test script | Developed in Javascript and usually be delivered along with automation test framework |
| Test data | Data used for executing test scripts, usually stored in database, external resource or directly in the code. Test data is usually delivered along with automation test framework |
| Test report | Sprint based automation reports and/or release based automation reports |
| Automation test guideline | [README.md](http://readme.md/) file |