# short description of the test approach

The best end-to-end test automation approach for software depends on the type of software being tested. Generally, a test automation approach should involve creating a test harness or framework, developing test scripts, and executing the tests.

1. Test Harness/Framework: A test harness or framework should be created that is tailored to the software being tested. This should include tools for test data management, test case management, test automation, and reporting.

2. Test Scripts: Once the test harness or framework is in place, test scripts should be developed. These scripts should be written in a language that is supported by the test harness or framework and should include verification points, assertions, and error handling.

3. Execution: Once the test scripts are written, they should be executed. This should be done either manually or through an automated process. Automated processes should be used if feasible.

4. Reporting: After the tests have been executed, the results should be reported. Reports should include the status of each test, any errors encountered, and a summary of the results.

## How to select tests for automation?

I will consider the following aspects when it comes to testing automation

1. Identify the business value of the test: Start by understanding the business value of the test. Tests with higher business value should take priority.

2. Identify risk and complexity: Consider the risk associated with a particular test, as well as its complexity. Tests with higher risk and complexity should take priority.

3. Frequency of use: Consider how often the test is used. Tests that are used more frequently should take priority.

4. Maintenance cost: Consider the cost of maintaining the test over time. Tests with lower maintenance costs should take priority.

5. Criticality: Consider the criticality of the test. Tests that are more critical to the system should take priority.

6. Reusability: Consider the potential for reusing the test. Tests that can be reused in multiple scenarios should take priority.

Based on the above factors, First, high-level requirements will be automated while prioritizing the regression test scenarios. Then it will be narrowed down to lower-level test automation