



TEST AUTOMATION ARCHITECTURE

Author :
Shefali Arora, Test Automation Architect

Executive Summary

Test automation frameworks provide a structure to the complete process of testing and increase the overall ROI of the SDLC. A test automation framework is a set of guidelines such as coding standards, test-data handling, object repository treatment etc. This when followed during automation scripting provide benefits such as increased code reuse, higher portability, reduced script and maintenance cost etc. Agama Solutions addresses your technical as well as financial concerns by offering quick and cost-effective test automation framework for application testing across diverse domains. Our test automation provides wide test coverage and helps you reduce time to market without sacrificing quality.

Test Automation Architecture

Is it mandatory to have a test automation framework? Can we have automation without a framework? The answer is yes! Just like we can have programs written in low level and machine language - difficult to understand, tedious to maintain and without re-usability. Just like in programming we shifted to high level and then object oriented languages to increase the overall efficiency, reusability and maintainability of our applications, similarly it's time we move from simple record and playback tools to a more sophisticated design and framework for our test automation.

A Test Automation Framework is a set of guidelines like coding standards, test-data handling , object repository treatment etc, which when followed during automation scripting produce beneficial outcomes like increased code re-usage, higher portability, reduced script maintenance cost etc. However, these are just guidelines and not rules; they are not mandatory and you can still script without following the guidelines. But you will miss out on the advantages of having a Framework.

There are different types of frameworks being used and implemented in the industry.

1. Linear/Record-Playback:

The tester can simply record the steps of a test case and add assertions and checkpoints to validate the actual results with the expected values and re-run the same scripts again and again. The biggest disadvantage in using a linear framework is that there is minimal reusability and in case of any minor changes to the application the maintenance is expensive.

2. Test Library Framework:

In this framework reusability is the focus and all libraries are created first and then one moves towards calling those library functions as and when needed. But this also has its own drawbacks. In order to create libraries testers need to be familiar with some programming language and more time has to be spent on planning and then creating the framework.

3. Data Driven Framework:

In this framework, while the Test Case logic resides in Test Scripts, the test data is separated and kept outside the Test Scripts. Test data is read from the external files (excel files, text files, CSV files, ODBC sources, DAO objects, ADO objects) and are loaded into the variables inside the Test Script. Variables are used both for input values and for

Test Automation Architecture

verification values. Test Scripts themselves are prepared either using Linear Scripting or Test Library Framework. The only disadvantage that this framework displays is that it takes more time to plan the tests and gather and create test data files.

4. Keyword-driven Framework:

The Keyword-driven or table-driven framework requires the development of data tables and keywords, independent of the test automation tool used to execute them. Tests can be designed with or without the application. In a keyword-driven test, the functionality of the application under test is documented in a table as well as in step-by-step instructions for each test. The problem with a keyword-driven framework is that it does not pay a lot of attention to data driving the tests and high level of automation expertise is required to implement this framework.

5. Hybrid Framework:

Here at Agama Solutions, we use what we refer to as a hybrid framework. We try and leverage the best from all frameworks and ensure that we can get the maximum ROI from our automation testing and hence can result in best quality applications.

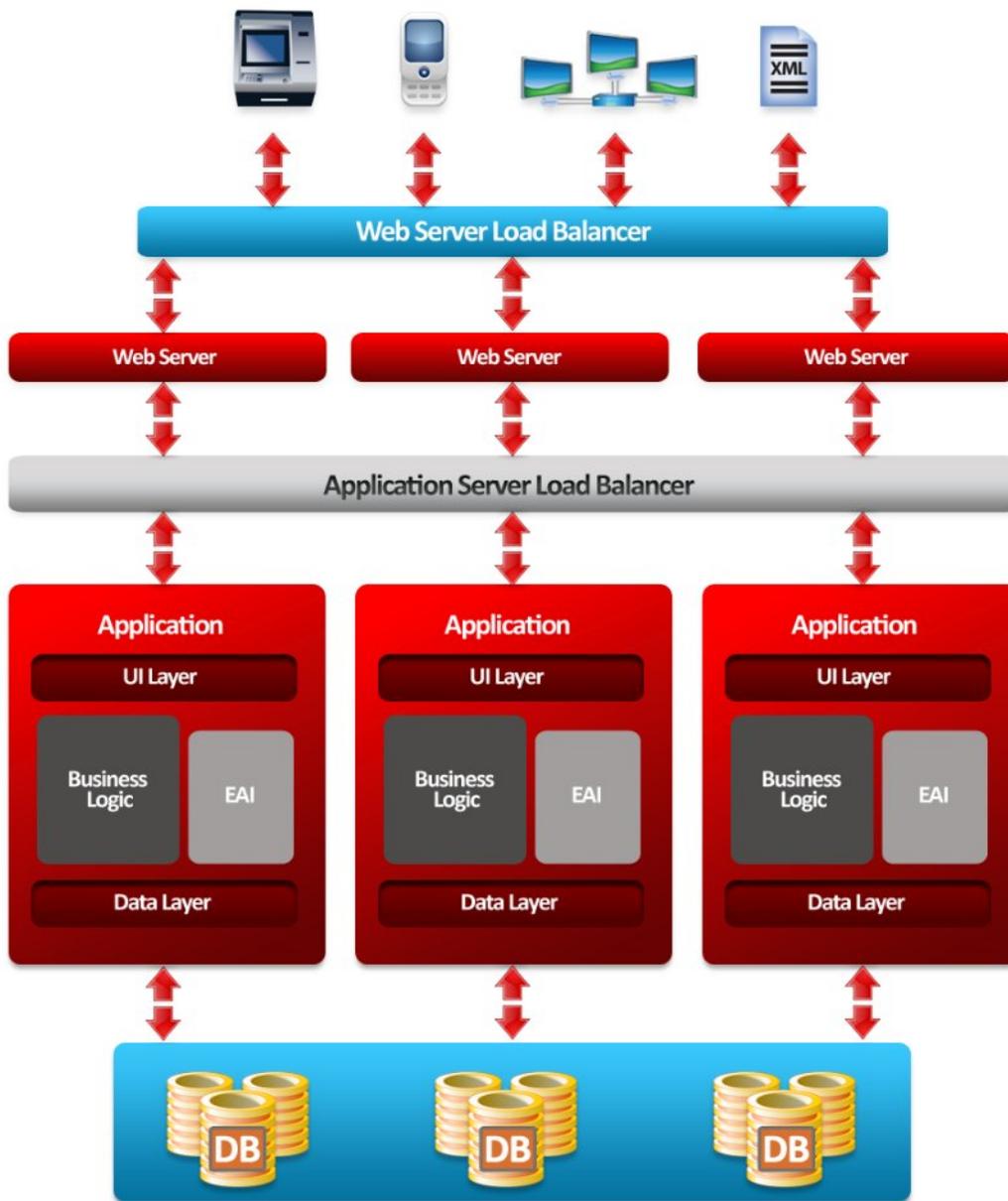
For creating a reusable, robust architecture that can be orchestrated according to your application we consider the following steps:

1. Understand the SUT and scope of testing

Before any real testing begins there's a lot of groundwork that needs to be done. The first and the foremost task in test automation is to understand the AUT, the architecture of the AUT, the methodology being followed and the scope of testing.

Most of the applications (enterprise or web based) have multiple tiers and complex architectures, so before any test automation tool or technique is finalized it is very important to understand the application.

FIGURE1: A TYPICAL N-TIER APPLICATION



2. Identify the testing needs

Understand the architecture of the application and plan the types of tests that will be performed on the SUT. Start from planning basic testing types like functional, regression, integration, system to more application and architecture specific types like web services testing, database testing, performance, security, compatibility, etc.

3. Identify the scope for automation

Once the QA team has a clear understanding of the application architecture and the requirements, the next important task is to identify the areas and the tests that can be automated. The main criteria to be considered while deciding on what to automate is the number of regression cycles to be run, the complexity of the tests and the criticality of the features. Simple routine tasks that are repeated very often are the first candidates for automation.

4. Evaluate Test Automation Tool

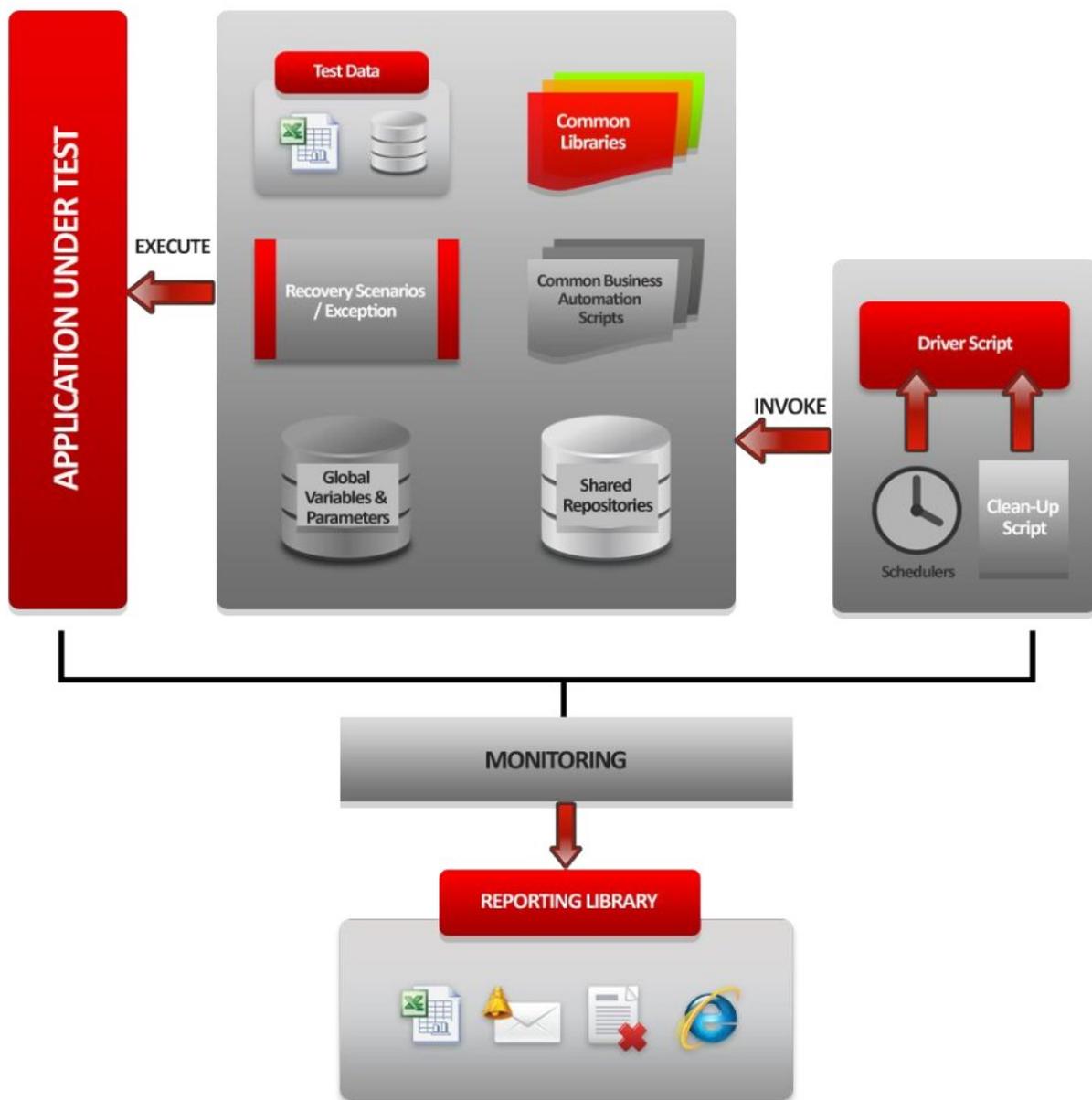
Once the needs have been identified the next important task is to select the right tool.



5. Design a Test Automation Framework

Automation Architecture

Once a clear understanding of the application and expectations is established, the next and the most important task is to build a framework. While designing a test automation framework reusability is the key. First try and find any reusable test wares (libraries, test data, recovery scenarios, etc) from previous projects that can be incorporated within the architecture.



- Begin with writing test scenarios and detail them in test cases/steps.
- Identify all repeating steps and create reusable function libraries.
- Gather or create test data necessary to facilitate the testing process.
- Create recovery scenarios and reusable exception handlers for all possible exceptions.
- Create driver scripts and clean up scripts to control the test execution.
- Schedule the tests for automatic execution, specially for light out testing.
- Export the results to understandable formats (html, excel or word).

Points to consider while creating the framework

- The framework should be robust.
- The test automation framework should be maintainable, as the application will undergo changes during the development process.
- The framework should be capable of handling dynamic events and objects.
- The framework should be capable to run in an unattended mode.
- The reports should be customized to contain only relevant data and should be automatically generated and emailed at the end of the run sessions.
- Create and establish best practices for the use and maintenance of the test framework.

6. Test Execution

A solid test framework foundation can lead to an overall impact on the test processes and can increase the overall quality of the application. Better testing would lead to a better quality application. Once the test framework has been built the driver scripts and the schedulers can be used to execute the tests as and when needed.

7. Reporting

Reporting capabilities of the tools play a big part in tool selection as well. Enhanced and customizable reports go a long way in helping the QA Team understand the overall execution status. The failures can also be described better with customized reports. The reports can be tool specific default report that the tool is generating itself or can be customized

html reports or XML reports that can be emailed to the relevant members of the team after the test execution finishes.

I would conclude by saying that build a framework that works for you following the best practices and tools. Ensure that you are not replacing your manual tests by automation or are neither are you going to have a solution for your problems that you faced with performing manual testing. Think of automation as an addition and not a replacement to manual testing and strike a balance between the two.

What makes Agama Solutions's Testing Services

We at Agama Solutions focus on the overall quality of our processes and products. With our testing experience and expertise in manual, automation, mobile, performance, web services and cloud based testing we have evolved our processes and frameworks over the years by following the industry best practices.

Agama Solutions's test automation framework is robust enough to be customized according to the customer and application requirements. The same automation framework can custom fit a heavy-on-content web-based application or an enterprise CRM or an ERP. The team of experts at Agama Solutions has the capability to convert test automation from an additional expense to a value provider that will add to the quality of the application.

Agama Solutions addresses both your technical and financial concerns by offering quick and cost-effective test automation that provides application testing for diverse domains. Our test automation provides wide test coverage and helps you reduce to market times without sacrificing quality.

Moreover, Agama Solutions offers a combination of onsite consulting and domestic outsourcing model for specialist services spanning across entire Software Test Life Cycle. Agama Solutions has a dedicated Quality Centre of Excellence QCoE built by leading successfully the testing projects over the years. Efficient processes, effective tools, core expertise, in-depth experience, dynamic management all in one place: These are some of the foundation elements of our QCoE.