



WWW.TECHNODIVINE.COM

BASICS OF AUTOMATION PART II

In continuation with part one of this article we would now be discussing following

- Automation Frameworks?
- Testing Process with Automation tools

Though, we have already covered “what the automation framework is?” I would still like to get a bit into the details so that it would make more sense and one would get clearer understanding of what this word mean and how important is this to understand the basics of what we are dealing with.

Even the most carefully planned and designed software, cannot possibly be free of defects and to find these defects is called software testing, which requires creating and executing many tests based on the results expected form the software.

Let us start from where we left in the previous article.

For Automation Geeks | By Bharat Kakkar

About Automation Frameworks

◆ What is a Framework?

- ❖ Framework is an abstraction in which solution providing generic functionality can be **selectively changed** by user code.
- ❖ In other words, it is a basic structure underlying a system, concept, or text. Framework is an essential supporting structure of an object and in our case the object is “Automation”

There is one more term that we need to understand well and that is “**Architecture**”. Think a bit, how you would respond to a question like “Describe the automation architecture and the framework used in your project?” do you know, what is the difference between “framework and the architecture” of the program. Let’s try to understand the same.

The dictionary meaning of word “Architecture” is the complex or carefully designed structure of something. Which means architecture of your automation/software project would define the conceptual structure and logical organization of all components within the project. These components would be the artifacts used for the program or automation execution. For an example the components of your project could be supporting libraries, third party controls (in case of development project), recovery scenarios (automation project), object repository (automation), pre-existing / reusable classes or objects (development project).

By now you must have a clear idea about how to define “architectural design” of a project. Let us now understand how the framework is different from the architecture of a project or let me say that let us restate what these terminologies mean and redefine these in simpler words.

So, a Framework is laid only after the architecture of a program / automation is been defined. This is because architecture defines that what all components would be required by the program and how these components are going to interact with each other. In lay man’s language, to build a house I would need a map to be created which would define how the structure will look like, it would also define the need of the end user and fixes the number of rooms, kitchens, bathrooms, doors, pathways and also defines their sizes as per the requirement. This is called architectural design, where in we define what kind of user is going to use the program and what the desired output or end goal is. We define what all modules will be required like reporting/ error handling / user input and so on and we also decide that how these components will be interacting with each other.

Once the architectural work is completed we start building the framework. We create the base on which our structure will be created. We do create the outline and the supporting structures like pillars, as in case aforementioned example of “building a house”. Building a framework is to materializing the ideas and the approach been decided in Architectural phase. In software world, it means creation of supporting libraries, classes, reporting module, error handling module or any artifact that will help in creating the end result. Any component which will be built as a part of framework will be reusable and independent in nature.

Please note once we are done with creating a framework of a house, as in case of previous example, user would not be able to build a wall outside the boundaries you have created within the framework. However, he may extend the framework if walls are to be built outside the boundary. Which means we are restricting a user not to change within the boundaries and on the same time we are leveraging the user to let the boundaries extended. Having said that I mean to say that framework does help users to be a symmetric and more efficient, while heading today's actual goal however, it doesn't restrict the user to extent the framework, if and when required. Let us take an example to better understand this, let's consider that a goal is to be achieved and we have n number of resources to achieve the goal. All of these resources would have their own set of thoughts and it would never be feasible to have symmetry in work, if a framework doesn't exist.

To summarize “Architecture” is a conceptual design of what is to be build and “Framework” is the implementation of the same concept to negate the variations and support the work towards the end goal.

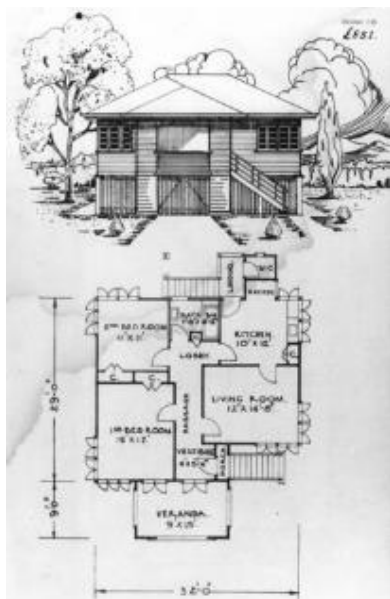


Figure 1 Architecture





Figure 2 Framework

Once the Framework is completed, we hand it over to the user so that he may customize the framework to his own need and built the desired output. This is up to the user now how he utilizes the framework to speed up the work and minimize the effort, because he already has enough resources to be utilized. Take an example of a task given to n number of employees to build a table and the only thing provided to them to start with is “raw wood”. The outcome of this would be some round tables, few square and may be triangular as well. Few of them would have four legs another may have three legs, until it is specified. However, if we provide them with a frame of a table with four legs and a square outline of the base. They would just have to cut the wood to fit the base and fix the legs, thereafter. In this case, even when no specification is given, the tables would be alike. Having said that means we have specified some rules without documenting the same, that’s what a Framework do.

Let us now see some of the real time implementation of different frameworks:

1. **.NET Framework** (pronounced dot net) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library known as Framework Class Library (FCL) and provides language interoperability.
2. **CUIT (Coded UI Test) Framework** is a thin layer developed on top of Microsoft Visual Studio Unit Test engine which helps reduce code, increases readability and maintainability, while also providing a bunch of cool features for the automation engineer.
3. **Selenium** is a well know open source testing framework, which is widely used for testing Web-based applications. It has different components and in that Webdriver has rendered the Selenium Remote Control obsolete, and is commonly referred to as Selenium 2.0.
4. **HP Unified Functional Testing (UFT)** software, formerly known as HP QuickTest Professional (QTP), provides functional and regression test automation framework for software applications and environments.

 The definition and meaning of “architecture” and “framework” would be common for a software program and automation project. Indeed, this would stand true for any object be it the automobile engineering for car manufacturing or construction of a building.

 Every tool has its own framework which is often referred to as a default framework of the tool.

Still Wondering! if these are the frameworks what is Keyword driven / data driven. Keep looking we will clear the same in next part of the same article.