**What is framework?**

Automation Framework is a set of guidelines and process you use to manage and optimize your automation testing. In simple words, Framework is a way to do the things in fashioned and organized manner! Framework is nothing but a concept.  
  
It's always possible to work without framework but you'll be doing a lot of rework, everything will be here & there and in most of the time you will end-up with a mess, especially in the large projects.  
  
Framework lets you predefine your automation plans, strategies and how you are going to proceed.

**Types of Framework:**

Framework can be of several types, but here are four main types of frameworks-

1. Linear scripting framework

2. Modular framework

3. Data driven framework

4. Keyword driven framework

5. Hybrid framework

But this is only theoretical. Practically, there is no 'Chinese Wall' between the framework types.

In every organization you work, you'll find that it's quite different from the other and not only on organization level, you can feel the difference within different projects!

Hybrid framework has no boundaries and open to take benefit of all possible type of frameworks.

1.Linear Scripting Framework:

It is the simplest of all Frameworks and also known as “Record &Playback". In this Framework , Tester manually records each step

( Navigation and User Inputs), Inserts Checkpoints in the first round ,then Plays back the recorded script in the subsequent rounds.

2. Data Driven Framework:

In this Framework, while 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, 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 Verification values. Test Scripts themselves are prepared either using Linear Scripting or Test Library Framework.

3. 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.

4.Hybrid Framework:

As the name suggests this framework is the combination of one or more frameworks. This hybrid test automation framework is what most frameworks evolve into overtime and multiple projects.

**How to design a framework?**

Practically, every application/project has its own expectation. You just need to start your work in organized way and believe me friends, at the end of the day; your framework will be ready automatically!!

You can't design a complete framework and then start automation! You start automation and framework will start building simultaneously.   
  
The first step of any test automation framework is to do a quick POC with few high priority test cases. By doing POC, you will identify the object identification issues, challenges, complexities, required test-data and limitation you have. Once you are ready with this information, you can proceed with next steps. On the basis of above information, you can decide what type of framework is most suitable to you.  
  
  
But before starting, you must take some time to decide your strategies about...

**1.ObjectIdentification   
2.Test-Data  
3.FunctionLibrary  
4.ErrorHandling  
5. Controller script**

**6. Documents and Manuals**

1. Object Identification:

One of the most important aspect of QTP automation is - How to identify objects i.e. using Object Repository (OR) or Descriptive programming (DP) or both. As per my understanding, don’t try to stick on one particular. Depending on the application/object types, mixed approach is most fruitful most of the time.  
  
Although  both OR and DP have their own pros n cons, but remember, you get full benefit of QTP only when you use OR like debugging, checkpoints etc. So make sure how to identify objects.

2. Test-Data:

How to manage you data? Data Table or Excel or may be flat/csv files.

3. Function Library:

Be clear about what type of function libraries you are going to use and why.

4. Error Handling:

What’s your plan to handle errors? Be clear about it.

5. Controller script:

Design a controller script, which will control all your automation. It should be neat. This helps to understand what actually you are going to call.

6. Documents and Manuals:

Keep all the documentation, user guide/manuals, keywords descriptions here.

Create different folder for these six things and keep data in folders accordingly.  
  
This was the one part.

Apart from this, from first day of your automation, keep only one thing in your mind... Follow general good practices!

Here are few points need to be take care in designing successful framework.

* Try to document each and every thing, as much as you can! like comments in script, modification etc.
* Don't stick with any particular object identification, use mix of OR and DP
* While using OR, don't use record and playback, just add object in the object repository and code the script manually.
* Follow proper Naming-Conventions for all your variables, functions, files.
* Define all the variables in you scripts, use Option Explicit.
* Try to keep you functions small to enhance reusability and portability.
* Never hard code any data in the scripts, use parameterization.
* Take regular back-up of your work.