**Use of framework in automation testing:**

Test automation: It is the use of software to execute tests and then determine whether the actual outcome and the predicted outcome are the same.

Uses:

* Quick and thorough testing of software
* Increase test coverage
* Reduce need of manual testing which is error prone and time consuming process
* Testing again and again build confidence about the new work added to the system did not break down and it is working
* Executing test can give us an idea about the quality of our product and implementation of the desired functionality
* Set of automation test suit can form a regression test suit. This is to make sure that functionality of the product did not change due to the changes done in the data
* Automation unit test suit help to find and solve the problem in early stages

Test automation using frameworks:

Framework: Systematic approach to perform testing

Testing framework: An execution environment for automated tests

Responsibilities:

* Defining the format in which the expectations to be expected
* Creating a mechanism in which test should be held on the application
* Executing the tests
* Reporting the results

Properties:

* It is application independent
* Easy to expand, maintain and perpetuate

Need of testing framework:

* Time needed for the tester to be productive, we use a framework which is application independent and also have the capability to expand with the requirements of each application.
* An organized framework helps in avoiding duplication of test cases automated across the application.
* Test frameworks help teams organize their test suits and in turn help improving the efficiency of testing.

Types:

* Modular Testing frameworks
* Data driven testing frameworks
* Keyword driven testing frameworks
* Hybrid Frameworks

Modular testing frameworks:

* It is built on the concept of abstraction
* Involves creation of independent scripts representing the modules of the application under tests. Which turn into large test cases
* Change made in one part doesn’t affect other parts

Disadvantage of modular testing framework: The main problem with modular frameworks is that the test script have test data embedded in them. So when the test data needs to be updated we need to change the code of the script. This becomes a big problem when the test script is large. For this purpose, data- driven testing frameworks have been introduced.

Data driven testing framework:

* In this test inputs and the expected outputs are stored in separate file (normally table format) so that a single driver script can execute all the test cases with multiple sets of data.
* Driver script contains navigation through the program, reading of the data files and logging of the test status information.

Advantages:

* Reduces number of test scripts needed to implement all the test cases
* Less amount of code required to generate all the test cases
* It offers great flexibility when it comes to maintaining and fixing bugs
* The test data can be created before test implementation is ready or even before the system to be tested is ready

Coded UI:

Automated tests that drive your application through its user interface (UI) are known as coded UI tests (CUITs). These tests include functional testing of the UI controls. They let you verify that the whole application, including its user interface, is functioning correctly. A coded UI test can automate the initial (F5) scenario, verifying that code churn does not impact the functionality of your application.

Coded UI Tests are particularly useful where there is validation or other logic in the user interface, for example in a web page. They are also frequently used to automate an existing manual test.

if you have a test case that was recorded in Microsoft Test Manager, you can generate code from that.