**Designing Test Automation Framework process:**

Creating **reusable**, **reliable**, **maintainable** Test Automaton Tools and Framework that can be used to build **reliable tests** **with great code coverage** and **align to the business needs** is significant aspect.

**Planning**:

* **Identifying** the **scope** of automation.
* **Extracting** the **automatable test cases** from **regression** or **smoke test suite**.
* **Identifying** the **different components** in the test automation scope. (ex: **Browsers, API services**, database, excel sheet, Interfaces (csv file, **xml file, json file** etc.).
* **Selecting** the **preferred programming language**. (ex: Java, C#).
* **Version Control System** where the **automation code** is **stored**. (ex: GitHub, Gitlab, SVN, etc.)
* Preferred **Unit Testing framework** (ex: Junit, TestNG, NUnit, XUnit etc.).
* Preferred **Build tools** (ex: Maven, ANT, Nuget, etc.,).
* Preferred **folder structure** (ex: QuickStart arche type).
* **Test Execution Mechanism** (ex: **Jenkins**, Windows Task Scheduler, command prompt).
* Preferred **Execution tools** (ex: Selenium Grid, Virtual machines, sacuelabs, headless).
* Preferred **Reports** (ex: Cucumber reports, TestNG reports, screenshots, etc.).
* Preferred **mapping mechanism** between **manual** & **automated test** (Excel sheet, Cucumber.)

**Designing**:

* Identify the **best set of tools** which **fits for company process**, budget, **application**, etc.
* **Create** a **skeleton project** with **best combinations** and **create automation test plan** document (Scope, Testing types, Environments, Test resources, Maintenance, Deliverables).
* **Review** the **document** with the **Development Manager** and **get sign off**.
* **Create** **Automated Testing task stories** in **JIRA**

**Implementation**:

* Create a **different levels of test suites** as Regression and smoke resource folder, feature files.
* Create a **Base Module** consists of all configuration which are required and commonly used by all Test Class and Page Classes (**BaseUITest** Class & **BasePageObject** Class).
* Create **Configuration Module** which holds config details in JSON which consists of DEV, SIT, CIT along with database details for data verification. Test run in **parallel** or **sequence** or will run in **local** or **grid**.
* **Test data Module** holds test data for testing purpose.
* **Page Module** Page Object Module **store all the pages** of the application. This is 1:1 mapping of Test Class and Page Class.
* Create a **Utility pkg** for most commonly used functions.

Types of Test Automation Frameworks:

1. **Linear** Scripting Framework: record and playback suing IDE.
2. **Modular** Framework: tester divide whole application in small modules.
3. **Data-Driven** Framework: test data is separated from the test scripts.
4. **Keyword-Driven** Framework: all actions are written in table format that we execute.
5. **Hybrid** Framework: combination of both.
6. **Behavior- Driven** Development Framework: which allows collaboration of teams.