**Automation Framework Design**

**Steps to succeed in Automation Design**

**A person walking up stairs

Description automatically generated**

* **Abstract:** creating a robust automation framework for our application, need to follow standardized rules and regulations.
* Good Automation Framework will guide our project to success, whereas bad framework will cost you money, time and resource wastage.

**Generation of Automation Framework**

**A person in a line of stages

Description automatically generated with medium confidence**

* **First Generation:** This refers to record and replay as published by different automation testing tools. This is our first generation considered as beginner level, majorly we focus non-reusable code where we difficult to maintain.
* **Second Generation:** Commonlywe use data driven framework it is to run same test cases varying test data. This test data can be stored in data tables or excels. The idea is to separate coding part form the data portion. It provides flexibility to tester modify the data easily rather than going to the code.
* **Third Generation:** Key word driven frameworks: As the name indicates hybrid needs mix of two or more frameworks, this is frequently implemented framework in testing phases. It could be modular approach. Separate set of data and driven by keywords.
* **Fourth Generation:** The key approach is to focus on building a model with specified input test data and estimated output results. Running the test cases after running compare actual with expected results with future of action which means it provides the execution results.
* **Fifth Generation:** Script less or code less frameworks: This approach is to use UI interface to record the objects build the test.

**Life Cycle of Automation Framework**

**A diagram of a process

Description automatically generated**

**What is automation:** Automation is to execute your existing regression test cases without any human intervention and your script should keep on updating stakeholders after every milestone. For good automation, manager and resources must plan, lot of things in advance, so that your automation suite is robust, maintainable and error free. All points which are taken into consideration before developing and automation suite is called automation framework.

Following are basic points you should keep in mind while creating automation framework.

* Proper documentation for running Automation suite.
* Easy to learn and easy to Execute.
* Easy to add new test Cases.
* Easy to add new Features/Product.
* Easy to maintain input data for different applications.
* Good Logging for each test case.
* Reporting to Stakeholders at each milestone.
* Scaling of your Automation Suite.
* Easy to increase/ decrease number of test cases in regression.
* Easy to increase/ decrease number of system.
* Can plugin script to check performance stats and Accuracy Stats.
* Automation suite setup in new machine.

A diagram of automation

Description automatically generated

Lifecycle of Automation, starts when, product in design phase and ends when product is obsolete and moved to rack. There are five different stages of automation, and each phase has its own importance. If you miss one phase than it will effect next phase and your automation framework as a whole.

**First Stage** is to understand the business and product. Identify tools to be used in framework. Test tool is decided by going through design document and requirement of the product. Factors like cost, availability of the tool and ease of integration with other tools will affect the decision. We can select tools internally used by the company, which can be embedded in the framework, for various task like logging, reporting, and integration with messaging apps like hipchat, slack, flowdock etc ...

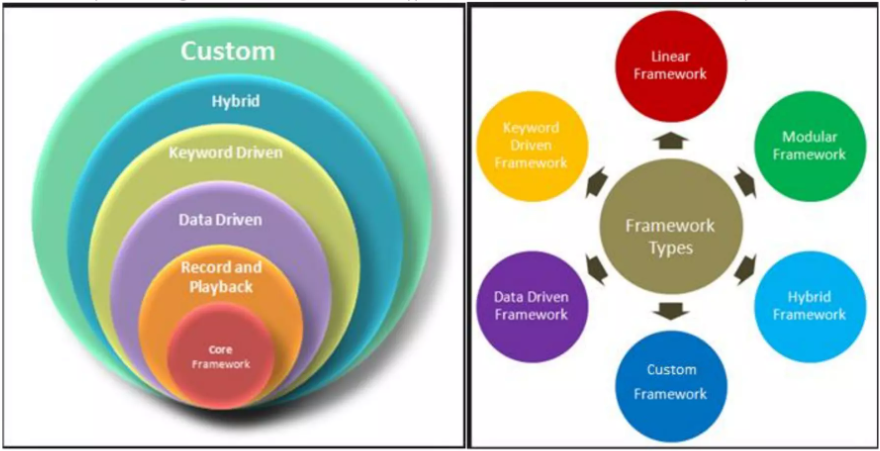
**Second Stage** is defining scope and coming up with the test cases you can and test cases which you cannot automate. Here you have to define scope and different milestones to track to your progress. Application Under Test, itself have different testing types to automate (namely Ul, End-to-End, server messages, logging, memory leak, threshold, load, stress, performance). Finalize scope before starting framework deign. Design should be flexible enough to incorporate different scenarios.

**Third Stage** is Planning, Designing and Developing your framework. Which is most time-consuming part of the whole cycle. This phase goes with the framework development. You plan and design your framework and as features are available you start automating those features. Separate code for features, logging, and reporting. If needed, you can easily add or remove features and test cases without changing framework and code. Config file is very important while implementing framework, you take necessary values from config files when your framework is running.

**Fourth Stage** is execution, and it goes side by side with third phase. Here you are developing new features and executing old features. You must analyze your test cases which are failed and understand the reason of failures, by going through logs. Reason could be an issue in application or flaw in your automation suite. You can embed a log parser for the test cases which are failed.

**Fifth Stage** is maintenance phase, when your product is stable and in the market and no more features are planned. You maintain your framework and scripts and update results of automation results on daily basis**.**

**Test Automation Framework Objective :**

****

* Test Automation Framework Objective. A test automation framework can be defined as a set of processes, standards and interactions between the components in which scripts are designed and executed. Different types of Automation framework can be developed.
* It is very important to define the objectives of a Test Automation Framework before starting its design and development. An organization may choose some or all of the objectives listed below, depending on its business needs. The design of the test automation framework should be based on its objectives defined by the organization.
  + Independent of Platform & OS
  + Support different testing types (UI, database, WebService)
  + Hybrid Framework is used to cater multiple needs (tables /keywords)
  + Ability to parameterize (recorded) scripts
  + Implement dynamic conditions & loop support
  + Test framework should be loosely /tightly coupled with AUT depending on business needs
  + Auto update of test results in a test management tool
  + Less dependence on highly skilled automation professionals
  + Ability to schedule and run prioritized test cases
  + Auto raise defect in defect tracking tool.
  + Data driven capabilities (input data, output data)
  + Test reporting objectives including email notifications.
  + End to End testing capability to cover business scenarios
  + Log defects in a defect management tool
* Test Automation Framework are guidelines, which when followed gives high quality results. Guidelines are like coding standards, test-input data handling, directory structure, logging, reporting etc. This helps in reducing code redundancy, higher portability, reduced script maintenance and better quality of framework

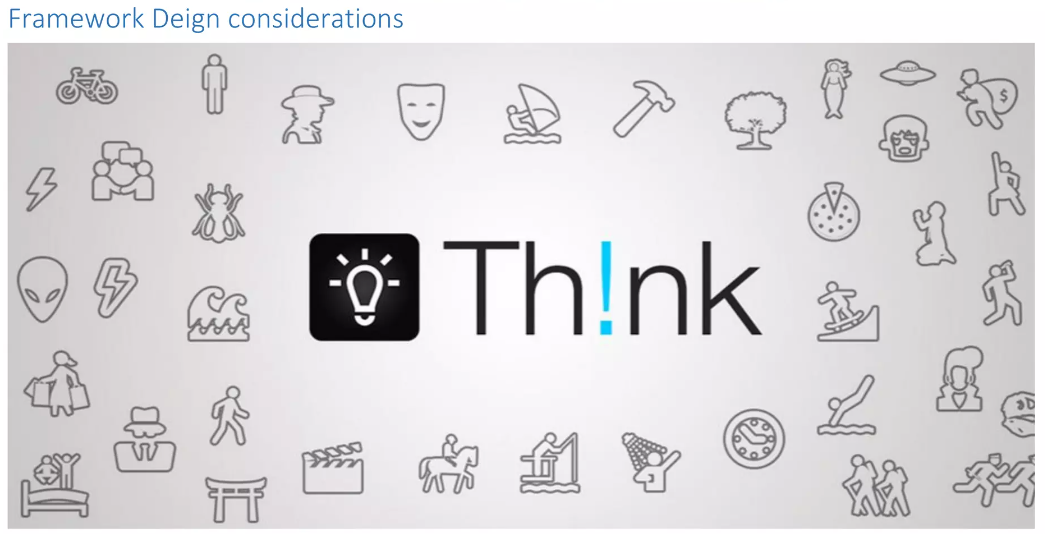
**Framework Methodology:**

Key parameters to keep in mind for framework design

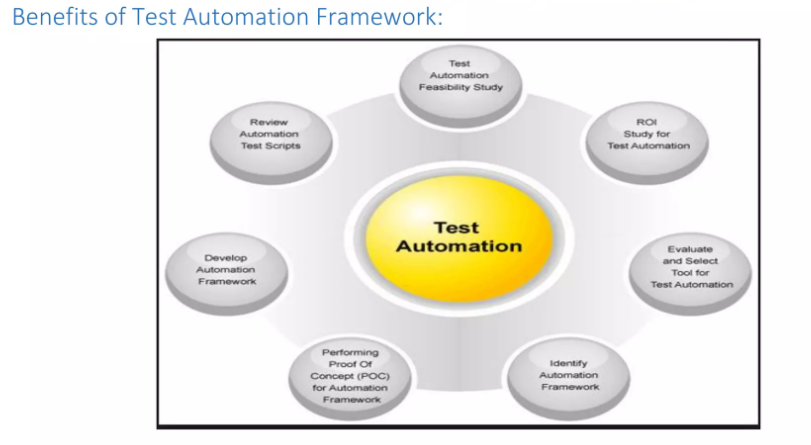
* Handle scripts and data separately
* Create libraries
* Follow coding standards
* Offer high extensibility
* Less maintenance
* Script/Framework version control
* Logging, Reporting to Stakeholders

**10 steps to Design good framework:**

* Identification of the Scope: Company oriented, Product oriented, Project Oriented.
* Identification of the Needs: Identify Types of testing e.g. Functional Test, Web Services etc. and application / modules to be tested, GUI testing.
* Identification of the Requirements: Find out the nature of requirements, identify type of actions for each requirement & identify high priority requirements.
* Evaluation of the Test Automation Tool: Evaluation checklist, Identify the candidate tools available in the market, Sample run, rate & select the tools, Implementation & Training
* Identification of the Actions to be automated: Actions, Validations & requirements supported by the Tool, messages between servers.
* Design of the Test Automation Framework: Framework guidelines, validations, Actions Involved, Systems involved, Tool Extensibility Support, Customs messages & UML Documentation.
* Design of the Input Data Bank: Types of Input file. Input files – Categorization & Design of file prototypes.
* Development of the Automation Framework: Development of script based upon framework design, Driver scripts, Worker Scripts, Record / Playback, Screen / Window / Transaction, Action / Keyword & Data Driven.
* Population of Input Data Bank: Different Types of data Input, Populate data from different data sources, Manual input of data and Parent – Child data hierarchy.
* Configuration of the Schedulers: Identify scheduler requirements & configure the schedulers



* **Type of framework:** Test Framework helps to organize test suite and improve test efficiency. Different framework focuses on different objectives.
  + **Data Driven:** Focuses on data management. If you need to execute your test cases with different data than this framework is suitable.
  + **Keyword Driven:** its runs test cases based on keywords. If you have many different scenarios for each features, you can implement this framework
  + **Modular:** If you want to separate coding of each feature than this framework is ideal. .It reduces redundancy and dependency of code.
  + **Hybrid:** This is used when someone have to implement more than one framework. This is widely used framework which can cater benefits of all framework.
* **Data Management:** Data is an important factor in framework deign. There could be different types of data set needed by you. User has to decide what to use for which data.
  + - Data for setting up automation suite
    - Data for the list of test cases
    - Data for the test cases, should run in automation suite.
    - Data for input to a script and expected result
    - Data for validation.
  + For configuration values, you can use text file
  + For validation and input/output result, you can use excel or csv are access.
  + For test cases, you can use excel or test case management tool
* **Generic Modules:** Test automation Framework have many reusable subroutines. If designed and used properly than it will increase the pace of framework development with minimal errors. These subroutines are defined at high level and based on the input to the subroutine it will do the necessary action. e.g. written a function to send mail after every milestone. Same function you can call at every milestone Log parsing: you wrote a function for parsing log based on search criteria and log path and output value. check(“SUCCESS”,”Feature3\_TC18”) OR check(“REQUEST”,”Feature3\_TC18”,3)
* **Rules of Framework:** This ensure consistency in your framework development. Defining set of rules needed for scripting and execution automation framework. It includes, coding, reusable function usage, directory structure to be followed, exception handling, creation of input data test, naming conventions for test cases and test scripts. It helps future development easy and streamlined.
* **Integration of tools:** Framework Design should allow easy integration of various scripts and tools you are planning to use. Integration with Email script, test case management tool or defect tracking tool. If automation suite finds any issue, than it should raise defect in defect tracking tool and update test cases management tool with the defect ID. And all stake holders should be informed about the action taken by the automation framework. Integration should not be dependent on the code.
* **Execution Configuration:** There are important things should be considered before executing your suite.
  + Feature selection
  + Test case selection for each feature.
  + Log level and Reporting settings
  + Milestone in your regression run
  + Stakeholders email ID
* **Reporting management:** Reporting is important part of framework and considered as a face of a framework. Reporting itself has many parameters to check.
  + - To whom it should go
    - Customizable and user-friendly report.
    - Different level of reporting Summary and Detailed
    - Format needed by stakeholders (html, excel, pdf)
* **Logging Mechanism:** Log generation is critical part of automation. It is vital while debugging issue and fixing the code. Logging can be done at different level, depending on configuration parameter

****

A well designed framework increases the speed and accuracy of the testing process, enables a higher return on investment (ROI), reduces maintenance and systematically minimizes risks. A well designed framework:

* Enhances efficiency in development of automated test scripts through modularized, reusable, and maintainable code and data.
* Provides a structured development methodology to ensure uniformity of design across multiple test scripts to reduce dependency on individual test-case developers.
* Reduces dependence on subject matter experts by automatically selecting the test to execute according to test scenarios and dynamically refining the test scope according to changes in the test strategy or conditions of the system under test
* Provides reliable issue detection and efficient root-cause analysis with minimum human intervention for the system under test
* Runs tests in parallel, runs on test machines in the grid or cloud, tests across multiple environments and application versions simultaneously
* Performs multiple levels and types of testing – functional, regression, load, performance, unit, integration etc.
* Eliminates deployment to multiple machines for testing different versions of an application.
* Is tool agnostic
* Runs in a continuous integration environment.

**Base Automation Design Structure:**

