Automation Test Plan

<please copy the text and paste in MS word to use this template for your project>

<please put comments on this document in case you have any suggestions.>

Document Control

Version	Date	Author	Change(s)
0.1 Draft			
0.2			

Sign off

Name	Title	Role	Status
	Test Manager	Owner	
	Project Manager	Sign off Required	
	Business Stakeholder	Reviewer Only	

1. Objective

This section specifies the objective of the automation test plan document and expectations stakeholders could have from this for a particular project.

2. Project overview

This section gives a general overview of the project to readers of the automation test plan. It helps the stakeholders to understand the project better and understand this document with respect to the project.

3. Assumptions

Specify any assumptions considered while drafting the automation plan. The assumptions need to be called out to the stakeholders so that they can understand the base understanding the automation test plan is based on. All assumptions affecting any phase of automation need to be called out.

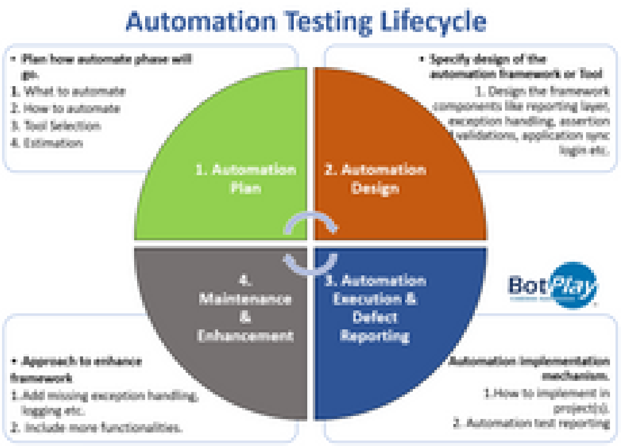
4. Return on Investment (ROI)

Not everything needs to or should be automated. Automation can be a costly affair and will give a negative ROI if the project is not a valid case of automation. Also, a project can experience negative ROI if automation is not planned properly.

This section should explain how the automation project will justify the investment verse if the same work is done manually throughout the life cycle of the product.
The formula to calculate ROI is -

ROI = Lifetime cost saved by automation - [Cost of developing automation + Lifetime cost of maintaining automation] + Lifetime cost of manual effort saved by automation

PS: For easy management of the automation testing phase, we can divide the automation phase into four major sub-phases to complete the automation life-cycle and can draw an automation testing plan around it.


Automation Testing Lifecycle
• Plan how automate phase will go.
1. What to automate
2. How to automate
3. Tool Selection
4. Estimation
1. Automation Plan
• Specify design of the automation framework or Tool
1. Design the framework components like reporting layer, exception handling, assertion validations, application sync login etc.
2. Automation Design
3. Automation Execution & Defect Reporting
Automation implementation mechanism.
1.How to implement in project(s).
2. Automation test reporting
4. Maintenance & Enhancement
• Approach to enhance framework
1.Add missing exception handling, logging etc.
2. Include more functionalities.
BotPlay

5. Automation Plan

This section defines what to automate and how automation will be planned. This section defines the foundation of automation planning. This section shall be covered the following items -

1. In-scope items / Out-scope items
2. Type of testing automation will cover
3. Automation approach according to layers of testing
4. Tool selection as per testing layers
5. Specify estimation techniques used

6. Automation design

This section provides details about how the automation implementation will be designed. In either case of in-house framework development or commercial tool implementation, the technical design of the automation implementation needs to be thoroughly reviewed so that framework can remain scalable and maintainable for its complete life cycle.

Not having a technical design document will lead to a low-quality framework because lack of -

- Unstructured modules/Monolithic application
- Improper design patterns
- Not following best coding practices
- No formal review processes
- Low code reusability
- No modularity at the function level
- No separate test phase of automation framework

7. Automation execution

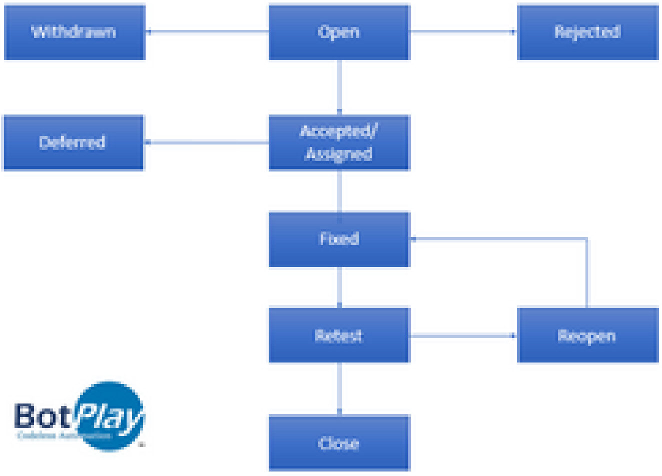
This section defines how automation will be implemented in the projects for automation test case creation and execution. Following considerations need to be called out in this section

1. How to implement at the project level?
2. What will be the branching strategy to manage automation code and test cases?
3. How will the test data be managed?
4. How will the test environment be managed?

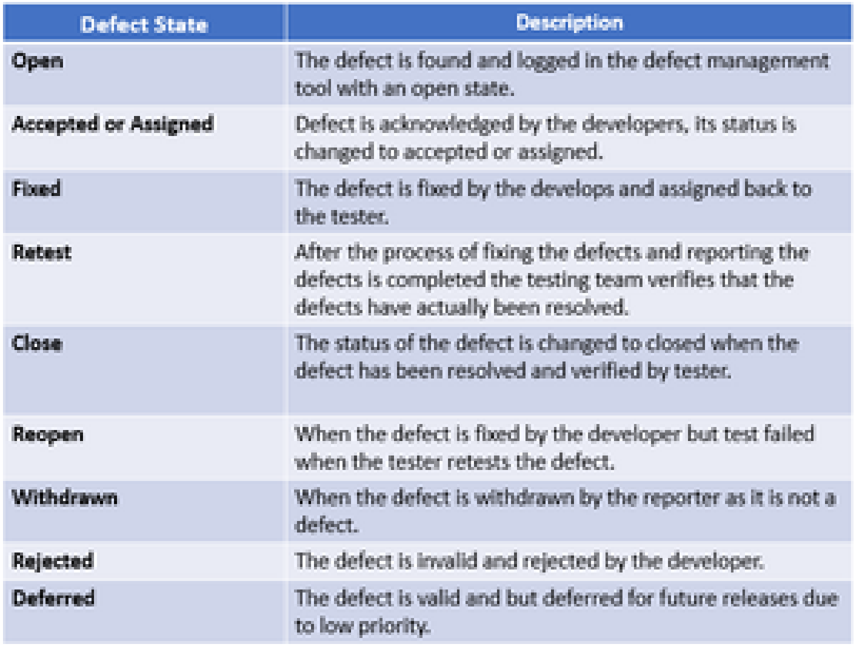
8. Defect Management

The ultimate purpose of automation is to maintain the quality of the product and raise defects when they are introduced in the code. It is essential to articulate how automation defects will be captured and reported in the defect management tool.

Defect Life-cycle
Defect Management life cycle contains the following stages:



Following is the description of the common states a defect can remain in its lifecycle.



Defect State	Description
Open	The defect is found and logged in the defect management tool with an open state.
Accepted or Assigned	Defect is acknowledged by the developers, its status is changed to accepted or assigned.
Fixed	The defect is fixed by the develops and assigned back to the tester.
Retest	After the process of fixing the defects and reporting the defects is completed the testing team verifies that the defects have actually been resolved.
Close	The status of the defect is changed to closed when the defect has been resolved and verified by tester.
Reopen	When the defect is fixed by the developer but test failed when the tester retests the defect.
Withdrawn	When the defect is withdrawn by the reporter as it is not a defect.
Rejected	The defect is invalid and rejected by the developer.
Deferred	The defect is valid and but deferred for future releases due to low priority.

9. Test Automation Reporting

The automation framework should be designed to produce a report that clearly articulates what had happened in the automation run. It should specify what are the test cases it ran, what validations were performed, what are the pass/fail test cases.

A good test report has the following components :

- Project Information includes the name and description of the project.
- Description of the test objective including the aim and enlisting the types of tests.
- Test Summary including the test case execution states.
- Defect report with the description, priority, severity, and status of the defect management process.

10. Automation Maintenance

Automation maintenance activities include -

1. Enhancing the expectation handling and error logging mechanism
2. Making framework more robust and scalable
3. Adding new functionalities in the framework to handle more use cases and test cases.
4. Code refactoring