# Automation Practices and Guidelines

## 1. Overview

This document outlines the best practices, challenges, and recommendations for automation in the Apply and Setup automation revamp project. The purpose is to ensure clarity, consistency, and efficiency in maintaining and executing automated test cases, with a focus on Q-Test tool management, UI vs. API testing balance, and addressing system test vs. SIT overlap.

## 2. Q-Test Structure and Automation Status

### 2.1 Folder Organization in Q-Test

The Apply and Setup folder in Q-Test should have the following subfolders, categorized by assets:

- DOCTA

- DOCTA in NAB1

- DOCTA in IB

- DCO

- DPL

Each subfolder will contain test cases relevant to the respective asset.

### 2.2 Automation Status Categories

Each test case in Q-Test must have its automation status updated under one of the following categories:  
1. Automated  
2. Partially Automated  
3. Not Automated

#### 2.2.1 Definitions

|  |  |
| --- | --- |
| Status | Definition |
| Automated | Test cases fully automated and executable without manual intervention. |
| Partially Automated | Test cases requiring manual updates (data, script, or configurations) before execution. |
| Not Automated | Test cases executed entirely manually, with no automation implemented. |

#### 2.2.2 Criteria for Automation Status

- \*\***Automated**:\*\* The test case should:  
 - Run seamlessly in a selected environment (e.g., SIT1, SIT4, SIT5).  
 - Generate all required data autonomously.  
 - Pass if the application environment is up unless there are defects or downtime issues.  
 - Allow occasional failures due to frequent application updates but not due to script instability.  
- \*\***Partially Automated**:\*\* Manual intervention (e.g., data sheet updates or minor script edits) is required before execution. This impacts the reusability of the test case in a pipeline environment.  
- \*\***Not Automated**:\*\* The test case is executed manually due to the absence of any automation, often due to system limitations or unstable scripts.

### 2.3 Using Q-Test Metrics for Improvement

It is essential to run queries in Q-Test to gather the following data for the Apply and Setup folder:  
- Total number of test cases fully automated.  
- Total number of test cases partially automated.  
- Total number of test cases not automated.  
This data helps in identifying areas of improvement and addressing gaps in automation.

### 2.4 Example: GreenPath Challenges

The GreenPath process requires a home loan setup in the OBP (Oracle Banking Platform). Challenges include:  
- OBP being a legacy system with no exposed APIs, requiring UI interactions for data creation.  
- Inconsistent script reliability (7/10 success rate).  
- Dependencies on customer registration and other systems.  
Proposals include engaging the OBP team to create APIs or highlighting the limitations as risks to stakeholders.

## 3. Challenges in UI Automation

### 3.1 Excessive UI in System Integration Testing

While UI testing is crucial for end-to-end flows, excessive reliance on UI for system integration testing leads to:  
- Flakiness due to frequent UI changes.  
- Slow feedback loops.  
- High maintenance costs.  
It is recommended to use API-level tests for stable functional coverage and only a minimal set of UI tests for critical end-to-end flows.

### 3.2 System Testing vs. SIT Overlap

The absence of a dedicated system testing phase leads to an overlap with SIT. Recommendations include:  
- Clearly separating system tests from system integration tests.  
- Using API-driven approaches for SIT to ensure faster, reliable execution.

**Final thoughts:**

While the Apply and Setup environment is highly complex, making it challenging to achieve 100% unattended automation, we can work towards addressing these issues to make progress. In the worst-case scenario, we can seek acceptance of these limitations from stakeholders, ensuring the current number of testing resources in the team remains unchanged.