<Company> Systems, Inc.

Intelligent Automation

Quality Assurance / Testing Cycle

Confidential

Table of Contents

[Introduction 3](#_Toc528243807)

[End to end Testing Implementation 3](#_Toc528243808)

[Best Practices 3](#_Toc528243809)

[Test Strategy 3](#_Toc528243810)

[Test Plan 3](#_Toc528243811)

[Test Case 4](#_Toc528243812)

[Defect Management 4](#_Toc528243813)

[Traceability 4](#_Toc528243814)

[Intelligent Automation Testing Approach 4](#_Toc528243815)

[Environment 5](#_Toc528243816)

[Dev Environment 5](#_Toc528243817)

[QA Environment 5](#_Toc528243818)

[Staging / UAT Environment 5](#_Toc528243819)

[Type of testing 5](#_Toc528243820)

[Functional Testing 6](#_Toc528243821)

[Integration Testing 6](#_Toc528243822)

[Regression 6](#_Toc528243823)

[Non functional testing 6](#_Toc528243824)

[Security/Vulnerability Testing 7](#_Toc528243825)

[Test Data 7](#_Toc528243826)

[QA Sign-Off Report 7](#_Toc528243827)

[Integrating other Intelligent Automation Technologies 7](#_Toc528243828)

[Appendix A – Defect log template 7](#_Toc528243829)

[Appendix B – QA Sign-off Report 8](#_Toc528243830)

[Appendix C – QA Test Strategy 8](#_Toc528243831)

[Appendix D – QA Test Plan 8](#_Toc528243832)

# Introduction

Intelligent Automation requires significant attention to ensure top quality outcome. It applies to targeted product/application and business process. It required set of testing/QA process, Testing type and environment that minimize errors and optimizes processes before Bot deployed in production.

# End to end Testing Implementation

The end to end testing of robotic process automation (RPA) implementation that will help <Company> benefit from less turnaround time, higher efficiency and better accuracy. End to end testing and QA offerings for <Company> must include a holistic strategy for RPA implementation as below.

1. Identify repetitive tasks performed by the business owner that can be optimized by implementing Intelligent Automation.
2. Identify the manual or automated testing requirements for each process.
3. Select those application/actions/output that are impacted or generated by the intelligent automation rather than testing entire process.
4. Enable end to end automation scripting by combining various automation tools with respect to integration.
5. Produce a single automation script from creating/inputting test data to Bot execution and validation across impacted applications/systems.
6. Reuse the automation script and test in higher environments like staging or UAT.

**Bot Development Life Cycle**

# Best Practices

## Test Strategy

It is a high level document which defines the approach for Intelligent Automation testing. It is basically derived from the <Company> Business Process Requirement document. Test strategy is developed by project manager or Business COE. It is kind of static document which sets the standards for testing so not updated often.

## Test Plan

Test plan is derived from <Company> Business Process requirement which is prepared by test lead or manager. The main goal of test plan is to include all the details related to testing such as what to test, when to test, how to test and who will be the tester. Test plan is often not updated but if there is some new feature or change is introduced then it has to be updated accordingly.

## Test Case

A Test Case is a set of conditions or variables under which a tester will determine whether a system under test satisfies <Company> Business Process Requirements or works correctly. The process of developing test cases can also help find problems in the Business Process Requirements or design of an application where Intelligent Automation communicates.

## Defect Management

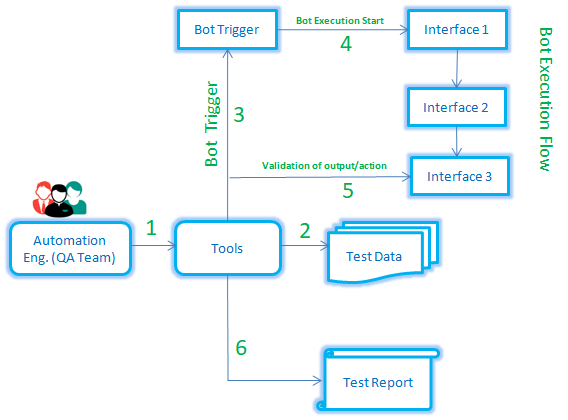
It is a process of improving quality and productivity by preventing the injection of defects into a Bot Development Cycle. It is virtually impossible to eliminate the defects altogether.

## Traceability

Traceability is the capability to trace relevant artefacts. In some cases, it is interpreted as the ability to verify the history, location, or application of an item by means of documented recorded identification.

# Intelligent Automation Testing Approach

Below snapshot shows high level flow of the how testing will be carried out for developed Intelligent Automation.



Above depicted automation testing approach is brief as below.

**Precondition check / test data creation**

1. Automation script is triggered by the QA team.

2. Automation script retrieves the test data.

**Bot Execution**

3. Automation script triggers the Bot through a schedule or from the central UI.

4. Intelligent Automation process is initiated and actions are performed by the sequence defined in the Intelligent Automation Bot.

**QA/Testing**

5. Bot automation validates output if it is successful.

6. Intelligent automation creates a test report for review.

# Environment

Setting up a right test environment ensures Bot testing success. A testing environment is a setup of software, hardware and network configuration to support testing for <Company> intelligent automation Bots. For test environment, key area to set up includes

* System and applications
* Test data
* Database server
* Front end running environment, etc

It is recommended to have below type of environments before Bot deployed to production.

Dev Environment Here is where the Bot designer develops, tests and checks whether the Bot runs successfully with the design. Once the developed Bot has been tested and the Bot developer feels that the Bot is working fine, the Bot then moves to the staging server.

QA Environment This environment is owned by test team. Once Bot designed then it pushed to this environment for full fledge testing carried out by testers. The testing team performs the different type of testing on this environment for developed Bot. Defects found in QA environment, gets logged into the defect management system. This type of testing on the QA environment is the final step before the developed Bot could be deployed on a production like environment. QA needs to sign-off in order to deploy it on the staging/UAT environment.

Staging / UAT Environment This environment is made to look exactly like the production server environment. The Bot is tested on the staging environment to check for reliability and to make sure it does not fail on the actual production server. This type of testing on the staging server is the final step before the Bot could be deployed on a production environment. The developed Bot needs to be approved in order to deploy it on the production server.

# Type of testing

Following are the different type of testing which will be performed during testing cycle.

Functional Testing contains all the functional specifications which form a basis for all the tests to be conducted as per <Company> Business requirements.

**Emulates Human Action:** Emulates human execution of the repetitive process using various application and systems.

**Perform Multiple Tasks:** Operates multiple and complex tasks across multiple systems. This helps to process transactions, manipulate data and send reports.

**Automated report generation:** Automates the extraction of data to come up with accurate, effective and timely reports.

**Information validation and auditing:** Resolves and cross-verify data between different systems to validate and check information to provide compliance and auditing outputs.

## Integration Testing

Integration testing is done to test the Bot/components/Interface when integrated to verify that they work as expected. This automation system can transfer data between disparate and legacy systems of the <Company>, by connecting them at the user interface level instead of developing new data infrastructure. It is required to test Bot integration between one Bot to other Bot as well integrated within Automation script.

**For example**

1. A developed Bot communicates with other Bot as part of integration to accomplish expected outcome.
2. Bot interacts with other set of system to fulfil expected Business process outcome.

Regression

It can be beneficial to QA processes which cover regression testing (which already exists in the <Company> production) and automating <Company> Business use case scenarios. It contains the information about how the system will be perceived from a <Company> business process perspective while Bot running.

**For example**

1. A developed Bot required the changes due to Business Process or Infra changes.
2. Deployed upgraded application where Bot is interacting with.

## Non functional testing

**Conduct high-volume repeated tasks:** Volume testing is a non-functional Performance Testing, where the Bot is expected running against to a huge volume of data. High volume testing is done to analyze the system performance by increasing the volume of data.

Intelligent automation can easily simulate rekeying of data from one system to another. It performs tasks like data entry, copying, and pasting.

**For example**

1. A developed Bot is referring file/database which contains high volume of data.
2. Multiple Bots are running and communicating data to Server (<Company> Kapaow server).

**Bot reliability:** Bot should be executed either infinite or finite long duration to ensure it produces expected outcome as per the <Company> business requirements.

## Security/Vulnerability Testing

Security/Vulnerability testing is a process intended to reveal flaws in the security mechanisms of an information system that protect data and maintain functionality as intended while Bot is running. Typical security requirements may include specific elements of confidentiality, integrity, authentication, availability, authorization and non-repudiation.

# Test Data

Test Data is created or provisioned by the <Company> Business team. It is strongly recommended to mimics the production sensitive data and in controlled manner. Developed Intelligent Automation script for <Company> business process retrieves the test data during execution. This test data will be used throughout testing cycle for Bot execution in different environment.

# QA Sign-Off Report

At the end of the QA cycle, QA team will produce the QA sign-off report which is an important deliverable. It will be prepared at the end of each QA Cycle. This report will contain the execution summary, defect summary and result of type of testing.

# Integrating other Intelligent Automation Technologies

While this document focuses on the RPA enabled automation deployments, it should be kept in mind that Intelligent Automation deployments may involve multiple other technologies including AI, ML, OCR, ChatBots or other specialized technology components. The COE Manager in their planning should ensure that all such Application Owners or other Technical Leads from each such area is included into the Testing lifecycle as per needs.

# Appendix A – Defect log template

The defect log template will help across team member to study/understand sufficient details about the defect found during QA cycle. It will also help to track towards to close the defect. It should be available and tracked in a defect management tools like JIRA, TFS.

Standard details to be entered by QA:

* Defect ID (Unique)
* Version
* Platform details
* Application/Interface
* Summary/Title
* Description
* Expected Behaviour
* Actual Behaviour
* Severity
* Priority (Must be entered by Business Owner)
* Supported Log, Snapshot

# Appendix B – QA Sign-off Report

QA sign-off report will help all stakeholders to stay in sync about the health of he developed Bot. It will also help to understand Bot execution impact on interface, infra and business process.

# Appendix C – QA Test Strategy

Following are the standard details which will be entered in QA test strategy report.

* Scope and objective: The objective of the business and how much testing scope is there is defined under test strategy.
* Business Issues: How much is the budget of the project, how much time is required for testing, how much resources are needed etc. are the part of business issues which needs to be considered before the actual testing starts.
* Testing approach: What type of testing is needed (performance, load, stress, functional etc.) and whether the testing is only manual or automation or both are some of the crucial points which define the testing approach.
* Test deliverables: What are the documents required from the testing team, how they would keep the record of the testing cycles etc. will be included here.
* Defect tracking approach: Which tool will be used for tracking the defects and how will the testing team communicate with the development team and how the flow would go for defects are decided at this point in test strategy.
* Training: If there is some complex or new tool is introduced in the business then it is helpful if the team members are given proper training. What type of training and the responsible person to conduct such training is defined here?
* Automation: If the project or business needs automation testing then the script language, tool used, reporting and code maintained is planned in test strategy.
* Risks: Nobody can anticipate all the risks beforehand but obvious risks can be avoided and also solution (if risk occurs) can be included in the document for future help.

# Appendix D – QA Test Plan

Following are the standard details which will be entered in QA test plan report.

* Test plan ID: This is a unique ID which defines the test plan. It can be a number or name or mix of both, as per the convenience.
* Test environment: This section defines what kind of environment is needed for the testing to carry out. For e.g. in device testing, usually a virtual set up is made to test emergency calling.
* Features to be tested/Not tested: This will have all the details about the features which tester needs to test and what are the feature which are not tested (may be because it is not yet implemented or not tested for that particular release).
* Entry/Exit criteria: These are the terms which define when to start or stop the testing. Standards will be defined under test strategy and followed by testers in test plan.
* Status: Whether a test case is passed or failed or not tested, all these test results are included in test plan with a proper reason.
* Types of testing: The types of testing required such as regression, functional, non-functional, stress etc. are defined and then executed by the respective tester.
* Brief Intro: Brief introduction is also included sometimes so that if any new member joins the team, he should get an idea how things work.