

Automating Quality Assurance with MuleSoft RPA and Composer

Case Study of a Component Manufacturing Company's Industry 4.0 Strategy:

In this case study we will outline how a component manufacturing company is utilising the "Hyperautomation of MuleSoft RPA with MuleSoft Composer" solution, developed by NJC Labs in order to automate their quality assurance process, as part of their larger strategy to implement Industry 4.0 technologies.

Challenge

The Quality Assurance (QA) team plays a crucial role in ensuring the components produced meet high-quality standards. Their daily tasks involve inspecting the components for various parameters with manual gauges. They then record these values, identify any defects, raise non-conformance issues, and prepare a comprehensive quality report.

Upon analysing the QA process, it became evident that, apart from testing, the other tasks involved were highly repetitive, rule-based, and data-intensive. As a result, the skilled workforce was being underutilised for tasks that were mundane and did not utilise their core competency of testing. This led to reduced overall efficiency, increased time consumption, potential employee fatigue, increased errors, and decreased productivity. In response to this, the Senior Management wanted to automate this process as part of their industry 4.0 implementation.

As a manufacturing company, they lacked a dedicated IT team or a specialised database. Nonetheless, the company planned to deploy the solution on a workstation and task their in-house maintenance team to oversee development, despite their limited IT experience.

Automating Quality Assurance with MuleSoft RPA and Composer

Objective

The objectives for a solution are highlighted below

- Automation of repetitive and mundane tasks.
- Streamline quality control processes.
- Faster turnaround time.
- Increased data accuracy.

Understanding the Process

Let us break the process in four steps:



1 Manual gauge testing of components

Quality inspector records data in an Excel Sheet



2 Spotting defects (Quality control)

Data that is outside the specification is identified.



3 Raise non-conformance

Escalate defective components to the Production Department.



4 Quality Report

Prepare a detailed Quality Report as per the product template.

Automating Quality Assurance with MuleSoft RPA and Composer

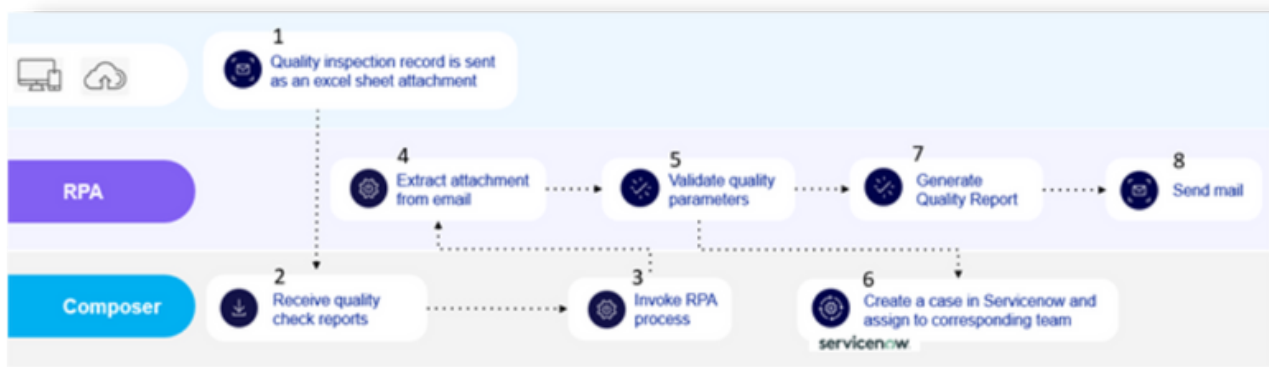
Solution

Below are listed the tools selected to match the company IT resource requirements.

- MuleSoft RPA – Can be deployed to the shop-floor workstation to automate manual repetitive tasks.
- MuleSoft Composer – A web-based tool used to integrate between email, excel and servicenow.
- ServiceNow – A web-based tool used to record non-conformance.

Integration codes developed in MuleSoft are easy to manage, understand and troubleshoot for anyone, including the shop-floor maintenance team. They are developed on the “no code, low code” ideology, allowing business users without any knowledge of coding to understand and easily use the solution provided.

In this solution, MuleSoft Composer and MuleSoft RPA work as members of an automation tag team as follows:



Automating Quality Assurance with MuleSoft RPA and Composer

Refer to the image on the previous page, the numbers below indicate what is happening at each part of the process.

1. Quality inspection record is sent to central quality mail box.
2. MuleSoft Composer initiates the subsequent business process automation when a new unread mail drops in the inbox.
3. MuleSoft Composer invokes RPA process "Read Data".
4. RPA bot extracts the email attachment and reads the data from Excel.
5. RPA bot runs the process of Quality Control by comparing the records against target specification ranges.
6. RPA bot sends back the specification data to the Composer to raise a case in ServiceNow.
7. RPA bot is programmed to generate a detailed Quality Report as per the different component catalogue product template.
8. RPA bot then completes the flow by sending this report back to the Quality Inspector's email id.

Automating Quality Assurance with MuleSoft RPA and Composer

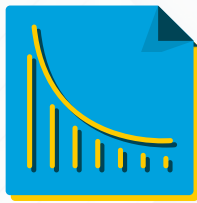
Results

The results of this simple integration are



2.5 X

INCREASE IN QUALITY
CHECKS FREQUENCY



70%

REDUCTION IN QUALITY
FALLOUT COSTS



15%

REDUCTION IN
OVERALL REJECTIONS



ZERO

ERRORS IN REPORTING
AND DATA ANALYTICS