Your Company

123 Your Street,

Your City,

ST 12345

(123) 456-7890

info@yourcompany.com

PROCESS DEFINITION DOCUMENT

<client name>

<process name>

Before any business process can be automated, it needs to be documented carefully.

The **Process Definition Document** describes how a specific set of operations is currently done in the organization, to collect all the information required in a clear, explicit, and shared way. What are the systems involved? Which steps are now performed manually to reach the end goal? What are the possible exceptions? How often is this process executed? What are the rules in place? This document must provide answers to these questions.

This information, once validated by all parties involved, will be used as the basis for the next phases in the automation of the process.

This document needs to be kept up to date with any possible changes in the involved systems and procedures for the whole lifecycle of the automation process, and all changes need to be signed off.

You can use this template to write great **Process Definition Documents** for your next RPA project. Just remember to delete all the example explanations, like this one! Let’s start!

## DOCUMENT VERSION HISTORY

Processes rarely stay the same for very long, and if the process changes or there is some new information, you should create a new version of this document, and archive the previous one. Remember to keep this section up to date!

|  |  |  |
| --- | --- | --- |
| **Version** | **Author** | **Date** |
| 1.x |  |  |

## 

## SIGNED OFF BY

When the process has been accurately described and all is good to go, make sure that the document is signed off by the relevant people, e.g. the process owner. By signing off this document, you ensure that they have carefully checked it through, and they agree to the content. This is important, so that later when there are changes in the process (it always happens, that's life!), everyone has a clear understanding of what the scope was originally.

|  |  |  |
| --- | --- | --- |
| **Name** | **Function** | **Responsibility** |
|  |  |  |
|  |  |  |

## 

## CONTRIBUTORS

Who are the people that have been involved in providing the information needed for this document? Who is responsible for it?

|  |  |  |
| --- | --- | --- |
| **Name** | **Function** | **Responsibility** |
|  |  |  |
|  |  |  |

# 

# Current process analysis

This section is where you will use your great analyst skills to describe how the client is handling the process now, before you move in and optimize and automate it like a pro. **What are the steps** that someone inside the company has to do each time to make this process work? **What software** do they have to use? **Where do they get the information** they need? **What happens** if something goes wrong? All these questions need to be answered in this section.

## High level description

Here you should give an overview of the process. **What** is this process for? **How often** do they run it? **When** do they run it? **Who runs** it? **What** are the steps?   
This should be crystal clear and not too long to read.

The process is used to....

The <employee> <each tuesday at 10 am> :

1. <Logs into the system>...
2. …
3. …

## Systems involved

Processes that are a good fit to be automated by RPA often span across different systems. One could be a web application, another a legacy system, or even just a spreadsheet. **List here the systems involved**, don’t forget to explain **what they are used for** and also if the user needs to have special authorization roles or rights, because the robot will probably need to have them as well!

|  |  |  |
| --- | --- | --- |
| **System** | **Used for** | **User role needed** |
| **<name of the system>** | <what it is used for> | <role and rights that the operator needs to have in the given system to be able to perform the action> |
|  |  |  |

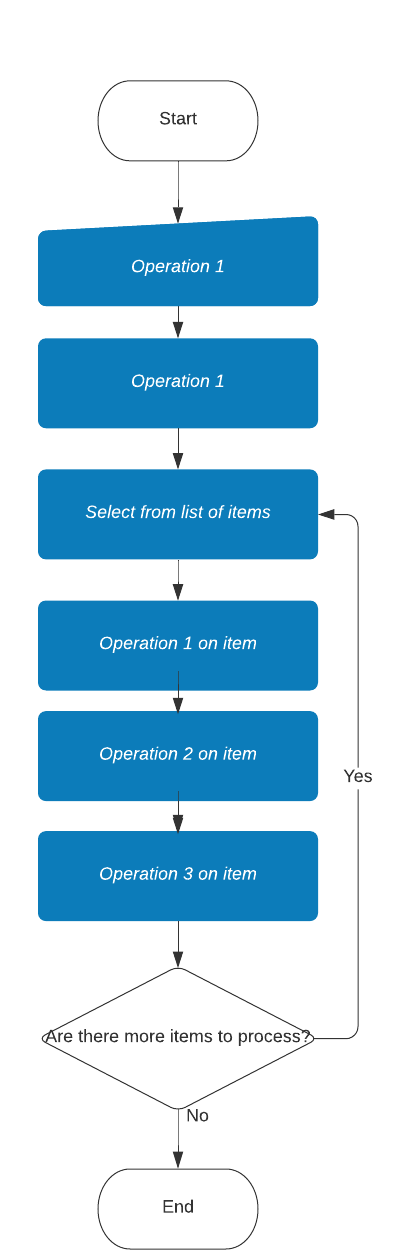
## 

## Process flow

A good flowchart is worth a thousand words! You can brush up on the (few) rules needed to make a good flowchart [here](https://en.wikipedia.org/wiki/Flowchart), and then get charting!

Here is a description of the process in the form of a flow chart:

Here’s what a very simple flowchart could look like:



## Detailed steps

### Download the orders file

<https://robotsparebinindustries.com/orders.csv>

### Enter the website and order

[RobotSpareBin Industries Inc. - Intranet](https://robotsparebinindustries.com/#/robot-order)

Click on button OK

Choose a head (dropbox)

Choose a body (select)

Choose the legs (type PN)

Type address

Click on order button

Print receipt and EMBED image as a PDF

### Enter the website and order

 create a ZIP archive of the PDF receipts

Store the archive in the output directory.

A good name for a step for example is in the format “<operator> <action> <object of the action>”. For example: **“The employee clicks the button "Add to Cart" next to the product. The product is added to the shopping cart".** This is not a hard rule though.

In the description of each step you are free to add anything that you think will help explain it better. Screenshots of the user interface, schema of the data involved, etc.

<Description of the step>

### <Descriptive name of step 2>

<Description of the step>

## Possible exceptions

### 

* The robot should complete all the orders even when there are technical failures with the robot order website.
* The robot should read some data from a local [vault](https://robocorp.com/docs/development-guide/variables-and-secrets/vault). In this case, **do not store sensitive data such as credentials in the vault**. The purpose is to verify that you know how to use the vault.
* The robot should use an [assistant](https://robocorp.com/docs/control-room/operating-assistants) to ask some input from the human user, and then use that input some way.
* The robot should be available in public [GitHub](https://github.com/) repository.
* Store the local vault file in the robot project repository so that it does not require manual setup.
* It should be possible to get the robot from the public GitHub repository and run it without manual setup.

### Logic Exceptions

Logic exceptions happen when something is wrong with the information that is being processed. For example, if an order has incomplete data, the operation has to stop. Or maybe the business has certain rules that it has decided on and the operator knows about: “do not sell more than 10 pieces a day for that product”. These need to be written down here carefully, because the robot will have to follow the same rules.

#### <Logic exception1>

<Description of logic exception1>

...

### System Exceptions

Software can have bugs, network connections can fail, passwords can no longer be valid: in all these cases we say that a **system exception** has happened. Write down all these possible cases, explaining what the operator sees, and also if there are ways to get around them.

#### <System exception1>

<Description of system exception1>

…

**And you are done**! Best of luck for your RPA project!