

# SOP for Manual Task Automation through CoE



## **Phase - 1: Agreement on Project**

- 1.1 Scope of Project
- 1.2 Work Out Clear Requirements
- 1.3 Sharing of Requirements: Alignment Meeting
- 1.4 Sharing of BOSCARD

## **Phase - 2: Allocation of Resources**

- 2.1 Selection of Project Manager
- 2.2 Selection of Automation Engineer
- 2.3 Alignment on IT Support
- 2.4 Technical Network Design

## **Phase - 3: Development Process**

- 3.1 Coding Phase
- 3.2 Local Test Phase/Alpha Test
- 3.3 Optimization Phase

## **Phase - 4: Test Phase**

- 4.1 Stakeholder Test Phase/Beta Test
- 4.2 Feedback Loop

## **Phase - 5: Rollout**

- 5.1 Installation of Python
- 5.2 Installation of Jupyter Notebook
- 5.3 Schedule Script on Windows Scheduler
- 5.4 Manual Run in Case of Scheduler Failure

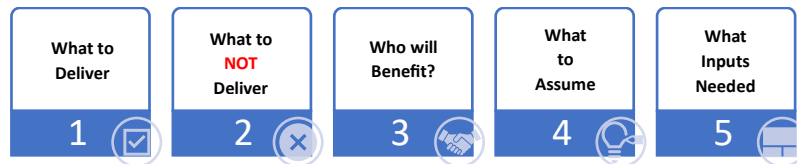
## **Phase - 6: Maintenance/Further Requirements**

- 6.1 Agreement on Changes
- 6.2 Allocation of Resources
- 6.3 Development Phase
- 6.4 Test Phase
- 6.5 Ideal Lead Time(s)
- 6.6 Rollout

## Phase - 01: Agreement on Project

### 1.1 Scope of Project

By scope it will be meant, what a project will perform; and more importantly what it will NOT deliver. This will be important to define clear boundaries around expectations. This will prevent potential disappointments and delays in the project lifecycle.



**Fig:** A clear definition of scope (will provide clarity on the 5 bullets above)

### 1.2 Work Out Clear Requirements

Clear requirements must include the following:

- How the task at hand is currently carried out, i.e., format and collection of IP formats, calculations
- How it will be carried out if automated, i.e., clear OP formats
- Communication channel of OP, i.e., email/MS Teams/WhatsApp/GSheet/shared folder
- To what extent the task will be automated

▲ Unclear requirements will pose roadblocks in the development of digital projects. When requirements are unclear, projects either won't start or stop mid-way, missing out on the objective of the undertaking.

### 1.3 Sharing of Requirements: Alignment Meeting

Requirements must be discussed and agreed upon formally by setting a meeting with all relevant stakeholders, LMs, and developers are present.

▲ A corridor walk or cafeteria gossip may not qualify as a requirement.

### 1.4 Sharing of BOSCARD

<b>Project Name</b>	
<i>mandatory</i>	
<b>Project Owner</b>	
<b>Project Stakeholders</b>	
<b>Project Type</b>	
<b>Project Team</b>	
<b>Approval From</b>	
<b>Launch Date</b>	
<b>Closed Date</b>	
<b>Risk (if any)</b>	
<b>Description (mandatory)</b>	
<b>Background</b>	

<b>Description (mandatory)</b>	
<b>Background</b>	
<b>Opportunity</b>	
<b>Scope</b>	
<b>Constraints</b>	
<b>Assumptions</b>	
Guideline	BOSCARD
IMAGE	CALCULATION

**Fig:** BOSCARD template

After the requirements are agreed on, a BOSCARD template will be issued by the developer(s) to the party desiring automation. This will clearly outline the assumptions, risks, constraints and opportunities of the project, once it is rolled out.

## Phase - 2: Allocation of Resources

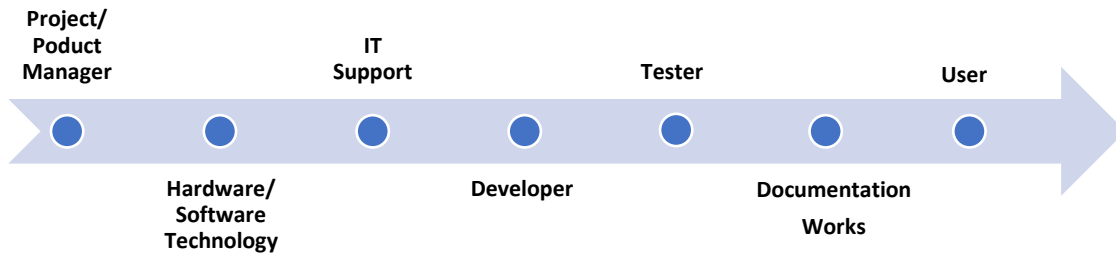


Fig: Resources for process automation

### 2.1 Selection of Project Manager

A project manager will be the automation product owner, often the prime user of the digital product in question. This person will stay in close collaboration with the developer during the development lifecycle. This person is expected to be the go-to person for the developer in case any confusion arises.

### 2.2 Selection of Scripter

Developers within the Supply Chain CoE come with a variety of skills. Hence, the initial steps are to:

- Determine if the task is internal/external-facing
- Determine if the task is analytical, or pertaining to gathering data from a multitude of sources
- Determine what type of data the task would analyze – text/image/structured/unstructured

Based on this, a developer with the right field of expertise will be assigned for the task by the head of CoE.

### 2.3 Alignment on IT Support

Respective function's IT Business Partner will be responsible for local/global IT related support such as access, cloud technology, recommendation of the right tool, etc. Also, function and respective ITBP will agree on the product owner. IT will be responsible for scaling and productionalizing digital deliveries.

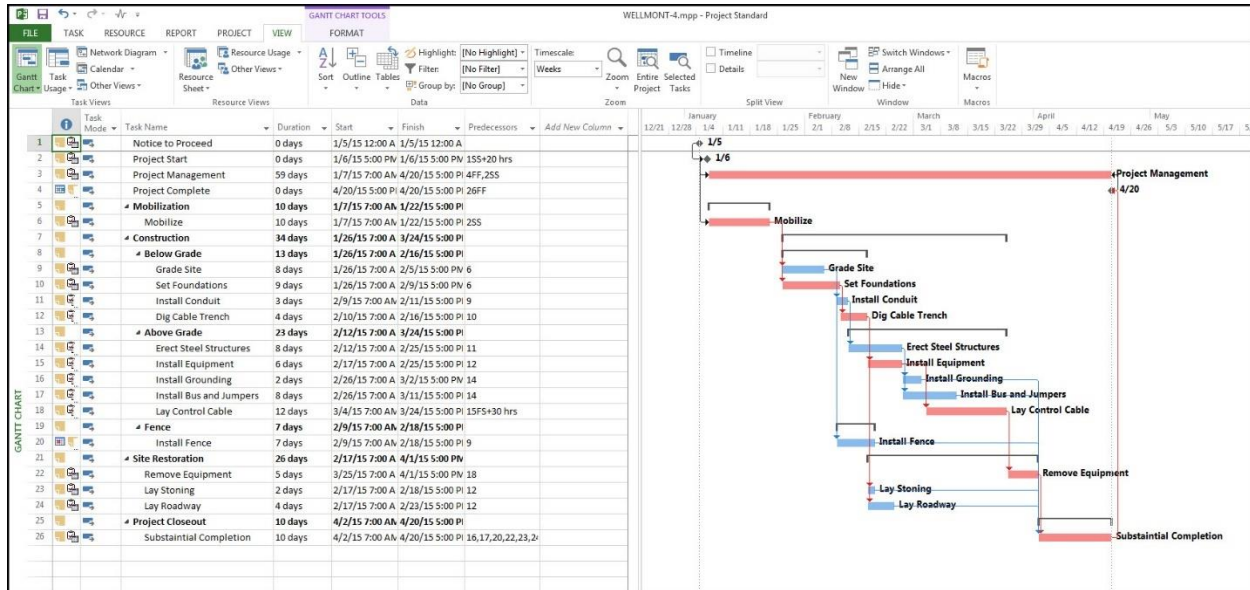
### 2.4 Technical Network Design

1	Projects under CoE	Timeline	Status	29.10.2023
2	Sl Project Name			Update
3	Ecom Portal		On Hold	Paused
4	Digital Skill Upgradation		WIP	UIPath botathon
5	DMS SCCF Loss Labeling	By October	WIP	Awaiting go-live
6	Neo	By October		
7				
8				1. New requirements in Order page initiated 2. Second phase development requirement fixed
9	QMIX Campaign		WIP	Deep dive Analysis
10	Eagle Eye		Done	
11	Value Stream Analysis		Done	Connected teams (Procurement, Network Analysis and Logistics)
12	PR Prioritization		WIP	Coding initiated, still in RnD
13				Connected with Phoenix team. Raised two tickets - 1. Response received on two tickets - no tools to bulk download files, weekly notification process being forwarded
14	11 Network Planning Master	By November	WIP	1. PACE team came back with response, HANA dump shared, they shared the discrepancies
15	12 RTVA Data Correction		WIP	

Fig: A simplified technical network template used for SC CoE

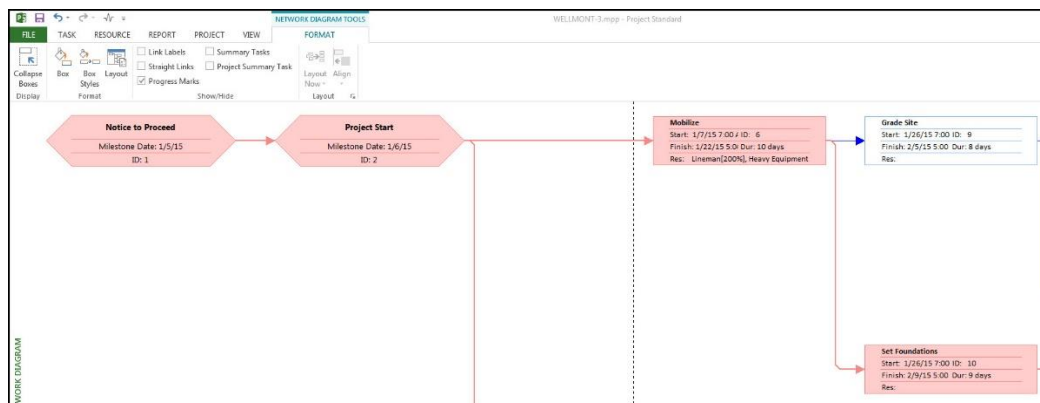
A technical network, in simple terms, will granularize the job into small task dependencies that will comprise the bigger picture. It can go as granular as the nitty-gritty of every other module the scripter develops. This will also record the dependencies, periodical status updates, timeframes and comments for effective tracking of the said project.

In formal terms, *A **project network diagram** is a graph that displays the order in which a project's activities are to be completed. Derived from the work breakdown structure, the terminal elements of a project are organized sequentially based on the relationship among them. It is typically drawn from left to right to reflect project chronology.*[\[source\]](#)

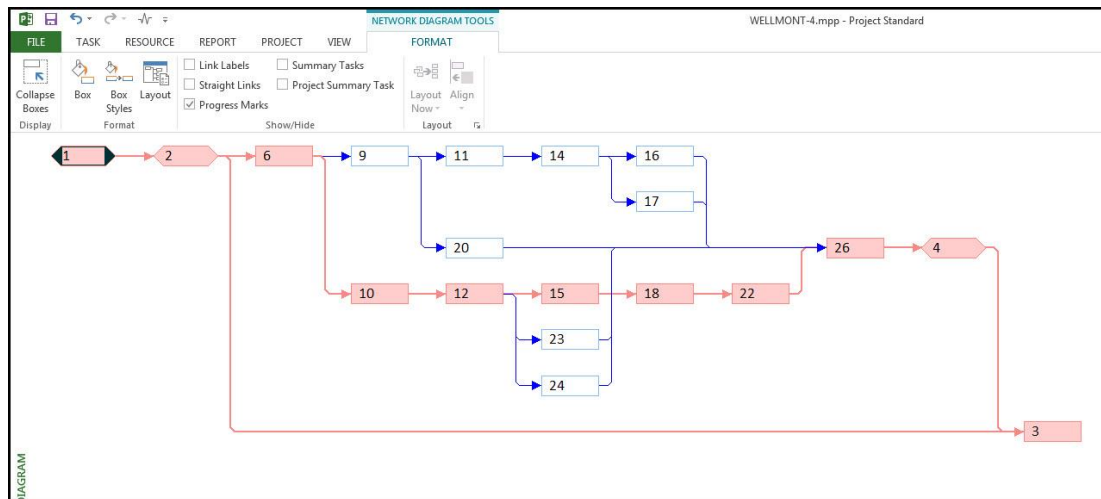


**Fig:** A proper project network on MS Project

Note that, tasks in red are along the **critical path** and tasks in blue are **non-critical**. Looking closer we see that some activities are performed in **parallel**, while others are in a strict **series relationship**. A network diagram will help us focus better on the **coordination** activity relationships.



**Fig:** A Project Network Diagram, with **milestones** (hexagonal), **task nodes** (rectangular), **critical tasks** (red), **non-critical tasks** (blue). A task node provides its **name**, **start**, **finish**, **duration**, **resources**, **ID**



**Fig:** Simplified project network diagram, displaying dependencies

With only the ID displayed as in the above, one can easily view the network more insightfully. The node (task) **predecessor and successor** are apparent for each node. The **critical path** is clearly visible. One can also distinguish the nodes or tasks performed in parallel and/or series.


## Phase - 3: Development Process


This phase will be the heart of the whole development lifecycle. In this phase, the real deliverables will be built. During this phase, a developer will need full concentration in scripting to come up with an amazing digital product expected to yield optimized output.

### 3.1 Coding Phase

This is the phase when workloads will be agreedly distributed among developers, and they will have full clarity of what to code. How a developer will develop the modules in the automation script will be at his/her discretion. Every developer is liberal to exercise their personal style of coding and it will be admissible as long as his/her code is offering agreeable readability.

### 3.2 Local Test Phase/Alpha Test

 LCs\_parsed\_test.xlsx  
33 KB

 64% of recipients have opened this mail.  
[See more insights](#) [Feedback](#)

Dear concern,

The service of automated parsing of datapoints from LCs is now developed and under testing. Please find (and verify) attached results from LCs shared [@ubl\\_tradealert](#).

Bank Name	LCs Received	LCs Parsed	LCs Incomplete	Incomplete LC Docs	Sec to Parse
HSBC	147	24	123	BDHS8C3K09286055-T02.PDF, BDHS8C3K09286016-T02.PDF, BDHS8C3K09282962-T02.PDF, ...	37
SCB	110	74	36	422010620483-AMD002-MT707-ADV.pdf, 422010619173-AMD001-MT707-ADV.pdf, 422010618520-AMD001-MT707-ADV.pdf, ...	69
BBL	185	2	183	F08102314517D001_SWIFT_071557613.pdf, 308523024708_SWIFT_190000397.pdf, 308523024708_SWIFT_172000653.pdf, ...	24

Note that, the statistics presented above reflect LCs shared since 26-Sep-2023. More banks will be added to the test eventually. This is an auto email via *win32com*.

**Fig:** Digital product in Alpha test phase

After the development phase, the Alpha test should take place. In this phase, testing will be done within the development team with the developer being present at the site. This phase will make sure the product passes initial screening, performed by LM and other CoE members.

### 3.3 Optimization Phase

During the Alpha test run, the developer will take note of the remarks and feedback passed onto them and will eventually incorporate those improvements into the program. This will make the delivery more robust. This window will also allow the developer to retrospect and work on improvements, which will finally improve the quality of the final delivery.

## Phase - 4: Test Phase

This phase will apply only to external stakeholders' requests. Internal-facing developments may move to maintenance right after alpha testing. The testing phase will ensure the usability of the data product in the context of end users and will ensure final goals meant to be achieved from the project are met.

### 4.1 Stakeholder Test Phase/Beta Test

At this stage, the output of the program authored will be shared with the product owner and end users on a periodic basis, with the developer being away. This will allow the stakeholders to assess the results and verify if the development has truly helped their purpose. In this phase, the end users will be allowed to share their feedback and developers will consent to the small amendments.

### 4.2 Feedback Loop

Stakeholder feedback at this stage may come in multiple rounds and small amendment initiatives may be undertaken, for greater comfort of use. However, this phase will not cater requests that may encumber a developer with a sizable additional development task on his/her plate.

## Phase - 5: Rollout

This is the final stage where the developed software will be handed over to the respective stakeholders. At this stage, the developer will have completed the development cycle and will officially sign off. At this stage, the project will mature enough to run without supervision. However, users are required to have certain data maturity for themselves to run the script independently.

### 5.1 Installation of Python

Since most automation scripts at Unilever will be written in Python, installation of this programming language is a prerequisite for undergoing transformation through automation. The following guide may be followed for Python installation:

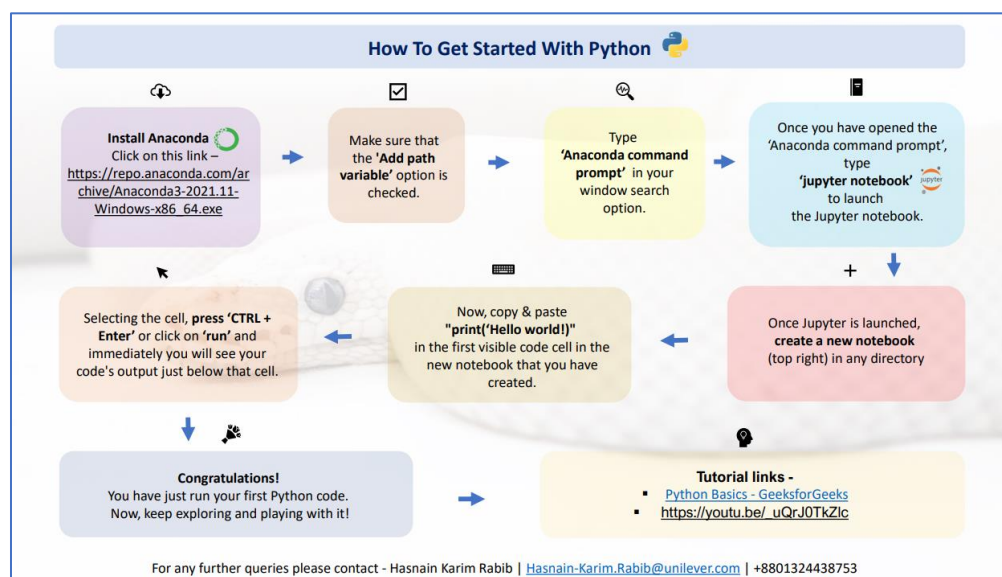
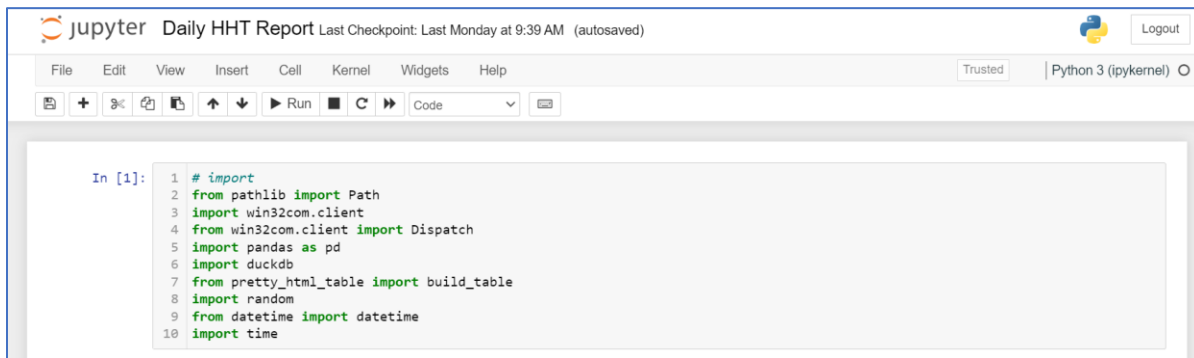


Fig: Python installation guide

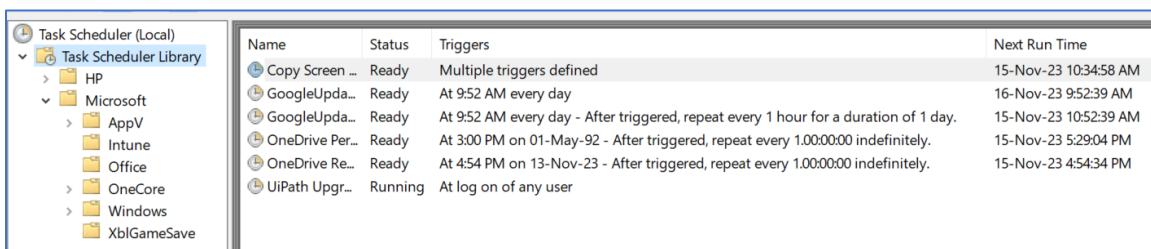
## 5.2 Installation of Jupyter Notebook



**Fig:** The cellular Jupyter Notebook IDE

Jupyter Notebook is an Integrated Development Environment (IDE) for Python program development. It is advantageous for its cellular provisions, a feature particularly helpful for people working with data. Hence this installation will equip the user with the power to write programs on his/her own.

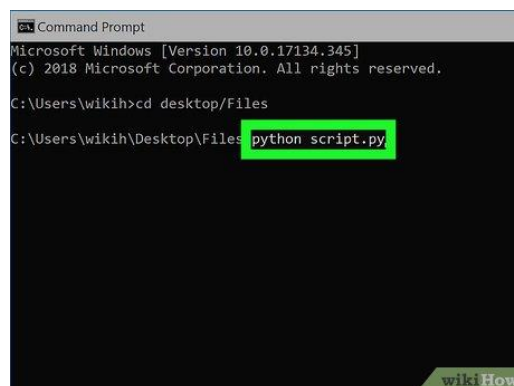
## 5.3 Schedule Script on Windows Scheduler



**Fig:** The Windows cronjob scheduler

Windows Task Scheduler is quite helpful in running the standalone scripts developed at Unilever. After Python installation, it will be helpful to run the scripts in the scheduler, for periodic updates of results.

## 5.4 Manual/Custom Run in Case of Scheduler Failure



**Fig:** Command to run Python program on demand

There will often arise a need to re-run a script or run a script outside of its usual schedules. Hence it is helpful to know how this is performed.



## Phase - 6: Maintenance/Timeframes

Any major changes will require the requester to go through this lifecycle once again. Also, all parties are advised to follow standard lead times mentioned in this section (6.6 Ideal ETDs), for each incremental step.

### 6.1 Agreement on Changes

### 6.2 Allocation of Resources

### 6.3 Development Phase

### 6.4 Test Phase

### 6.5 Rollout

### 6.6 Ideal ETDs

Task	Subtask	Standard Lead Time
Agreement on Project	Define scope	3 days
	Chalk out clear requirements	2 weeks
	Alignment meeting	2 sittings
	BOSCARD fillup	1 week
Allocation of Resources	IT support alignment	subject to discussion
	Project network design	continuous process
Development Process	Coding phase	several weeks/days, as per complexity
	Alpha test	2 weeks
	Optimization phase	parallel to Alpha test
Test Phase	Beta test	3 weeks
	Feedback loop	parallel to Beta test
Rollout	Installation of Python	2 weeks (including installation and scheduling) +
	Installation of Jupyter Notebook	
	Schedule script	
Maintenance	Documentation	subject to task complexity
	Software updates	
	Further changes	

**Table:** Each task with their respective estimated times of delivery (ETD)

## Appendix

To be added ...

## Prepared By

Shithi Maitra, Asst. Manager - Customer Service Excellence

## Revised By

Mehedi Asif, Sr. Manager - Customer Service Excellence