



Prodapt powering
global telecom

**Increasing the pace of your process
automation programs**

Credits

Rajeshkhanna J

Manjunath C

Sarvagya Nayak

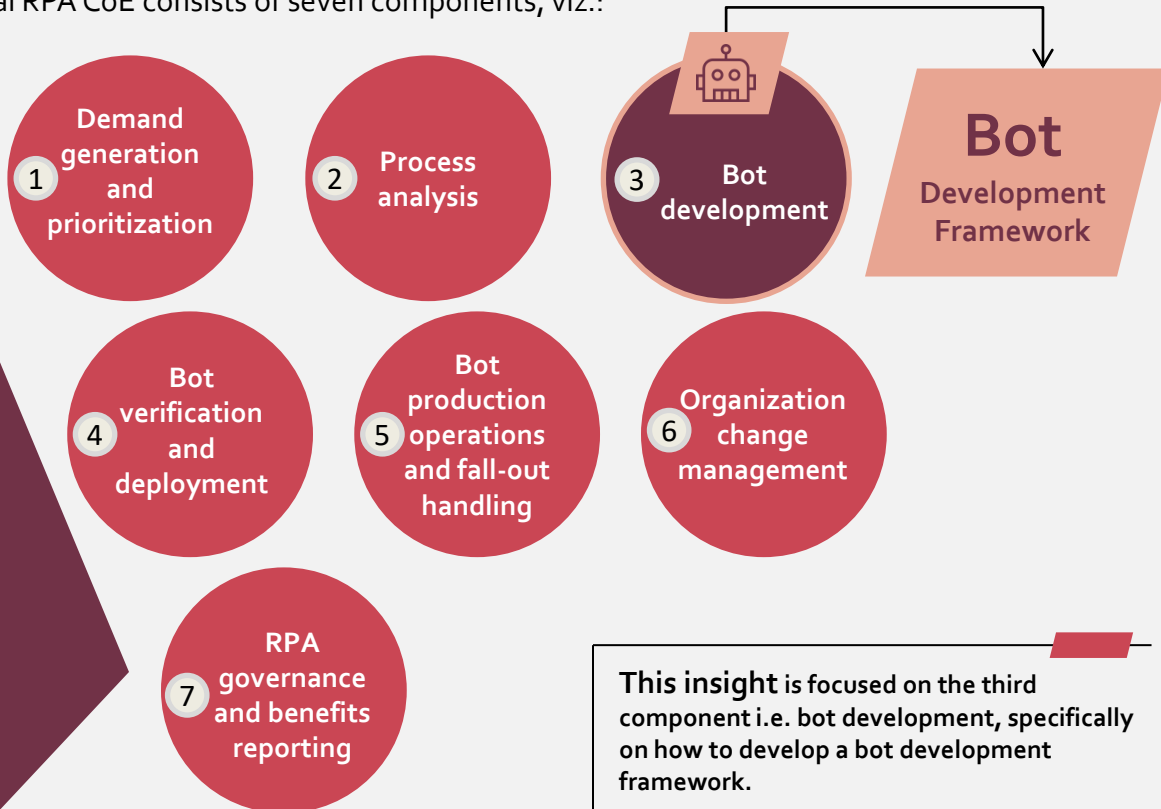
RPA Centre of Excellence (CoE) – A definitive answer to DSPs' scalability challenges

DSPs across the globe are leveraging robotic process automation (RPA) to increase their operational efficiency and are at various stages in their journey.

Most of them have at least tested waters with a PoC, if not the actual implementation. But scaling RPA and making it an organization-wide success is a big challenge.

One definite answer to this is to develop a robust Centre of Excellence, which not only defines the best practices but also strives to achieve the organization goal.

An ideal RPA CoE consists of seven components, viz.:



Bot development framework – A reusable template to develop bots

Bot Development Framework

- Bot development framework is a methodology, which standardizes bot development throughout the organization
- It is a template or skeleton providing generic functionality and can be selectively changed by additional user-written code
- It adheres to the design and development guidelines defined by the CoE, performs testing, and provides application access
- This speeds up the development process and makes it simple and convenient enough for business units to create bots with no or minimum help from RPA team.
- It helps saving time in development, testing, building, deploying and execution

Steps involved in developing a bot development framework:



CHOICE OF
DATA FLOW
REPRESENTATION
LAYOUT



SETTING UP
CONFIGURATION
FILE



BUILDING
CREDENTIAL
MANAGER



DEFINING
PROCESS
EXECUTION
STEPS FOR BOTS



ROBUST
EXCEPTION
HANDLING AND
APPLICATION
RECOVERY



LOGGING

Choosing the appropriate representation layout for the framework and setting up configuration file to store common data



CHOICE OF DATA FLOW REPRESENTATION LAYOUT

In any bot studio, there are primarily **3 types** of layouts available:

Linear | for simplest processes, which do not include decision making but only navigation.

Flowchart | for complex processes with multiple branching into logical operators.

Hybrid | helps to keep the process running and provide cleaner routes for various circumstances and is composed of various states.

Recommendations

Use a hybrid layout instead of linear or flowchart in the bot studio. This addresses complex process flows while allowing independent flowcharts to be accommodated within the master layout.



SETTING UP CONFIGURATION FILE FOR COMMON DATA

Common data such as application URLs, orchestrator queue names, maximum retry numbers, time out values, asset names etc. are prone to updates frequently.

Recommendations

- **Configuration file** - Create a "configuration file" to store these data in a centralized location. This will increase process efficiency by saving the time needed to access multiple applications. E.g.: Define the number of retries to be attempted in case of an application exception.
- **Pre-defined webpages** - It should also have pre-defined webpages with business logic defining how to launch/terminate the application and steps to be performed in it.
- Process specific coding should also be done inside these pre-defined pages.

Building credential manager to store login information and defining the execution path



BUILDING CREDENTIAL MANAGER

For bots to access multiple applications, it is important that they have easy access to credentials while maintaining their security

Recommendations

- To facilitate this smooth logging-in, the framework should have a credential manager which serves as a secured and centralized credential repository
- Only an credential management admin, having master key to the credential manager, should be allowed to create credentials in the form of credential keys
- These credentials should be encrypted using industry standard FIPS 140-2 approved cryptographic module with AES-256-bit encryption and can be accessed by authorized and authenticated users only

DEFINING PROCESS EXECUTION STEPS FOR BOTS



Initiate
State

Get
Transaction
Data State

Process
Transaction
State

End
Process
State

Recommendations

- Open configuration file & required applications at the initiate state itself
- Use 'wait stage' after any stage that causes screen update. It not only absorbs any latency in the process but also makes it fast.
- All queue items should terminate either with the status 'completed' or 'exception'
- The reason of exception at sub-page, in maximum cases, can be 'network time-out'. Retry such exceptions, as per the limit defined in configuration file, before sending it for exception-handling.

Defining the exception handling rules and logging policies of end-to-end process



ROBUST EXCEPTION HANDLING & APPLICATION RECOVERY

Having a robust exception handling mechanism and transaction retry scheme is the backbone of efficiency of this framework.

Recommendations

- Categorize and capture exceptions as either application exception or business exception
- In case of application exception, define the maximum number of retries
- Do not retry business exception
- On top of saving the textual log message of the error, it is also useful to take a screenshot of the triggered exception in order to better understand it



LOGGING

Log messages are very important since they offer a report of what has happened.

Recommendations

It is recommended to perform logging:

- At the beginning and at the end of every workflow
- When data is coming in from external sources
- Every time an exception is caught at the highest level
- When a transaction is successfully processed or failed with application or business exception
- Each time an application initialization is retried
- While opening an application, closing an application and killing a process
- While capturing and saving exception screenshot

Log messages should describe the nature and scope of current state clearly. The framework should have log message templates for each state.

Key takeaways and benefits



SAVING TIME

It saves development, testing and deployment time by removing the need to develop the bot from scratch.



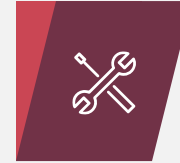
SIMPLICITY

This framework simplifies heavy and complex bot designing, thus making bot configuration easy for any business unit.



REUSABILITY

It allows for easy plug and play of varied applications and data sources with the bot as per the process requirement.



EASE OF OPERATIONS

Since all the bots designed in an organization follow the same protocols and guidelines, it becomes easier for operations team to maintain, troubleshoot, reconfigure or reuse bots.

Bot framework helps in reducing development, testing, deployment and review time by **30%**

Get in touch

USA

Prodapt North America
Tualatin: 7565 SW Mohawk St.,
Phone: +1 503 636 3737

Dallas: 222 W. Las Colinas Blvd., Irving
Phone: +1 972 201 9009

New York: 1 Bridge Street, Irvington
Phone: +1 646 403 8158

CANADA

Prodapt Canada Inc.
Vancouver: 777, Hornby Street,
Suite 600, BC V6Z 1S4

UK

Prodapt (UK) Limited
Reading: Davidson House,
The Forbury, RG1 3EU
Phone: +44 (0) 11 8900 1068

EUROPE

Prodapt Solutions Europe
Amsterdam: Zekeringstraat 17A, 1014 BM
Phone: +31 (0) 20 4895711

Prodapt Consulting BV
Rijswijk: De Bruyn Kopsstraat 14
Phone: +31 (0) 70 4140722

Prodapt Germany GmbH
Aschheim: Sonnenstraße 31, 85609
Germany

SOUTH AFRICA

Prodapt SA (Pty) Ltd.
Johannesburg: No. 3,
3rd Avenue, Rivonia
Phone: +27 (0) 11 259 4000

INDIA

Prodapt Solutions Pvt. Ltd.
Chennai: Prince Infocity II, OMR
Phone: +91 44 4903 3000

“Chennai One” SEZ, Thoraipakkam
Phone: +91 44 4230 2300

Bangalore: “CareerNet Campus”
2nd floor, No. 53, Devarabisana Halli,
Phone: +91 44 4903 3000

THANK YOU!