Building and deliverying RPA solutions though very exciting but has its challenges especially in production environment. As Post production or after go-live is has many ‘unknowns’ which could impact the overall effictiveness of automation.

For development and design there are many best practices but here want to share few learnings which would help during the go-live / production support phase:

1. WorkQueues (Blue Prism):

Proper design around the work queue can significantly improve the Operational efficiency in produciton environment. Lets say a simple process of reading from excel input and applying those pn an application needs to be automated. One way to achieve this is to read the rows one-by-one from excel file update the application and update the proecssing status row-by-row. But this approach will cause the bot to run unnecessary longer due to frequent I/O operations while updating the status of the row being processed. Also if one has to track the number of rows then extra development effort is required.

Using the WorkQueue can signficantly help in this case as they provide important features like inbuilt screen in Blue Prism Control Room which shows all the rows loaded, their processing status etc. Most important I feel is the feature of one can easily re-process failed rows - by changing the status to **Pending**. This helps in **re-running only failed items** after taking proper corrective actions related to configuration, setup of the Robot or Application etc.

1. Email Reports and Alerts:

Many Bots generate reports or update status on excel sheet for transaction based automation process. General tendency is to send such reports as **attachments**.

I generally prefer to save these reports on common drive or portal and **email body should only have the full path** to the business users. This practice helps avoiding any unwanted data leakage issues. Only exception is when business user wants reports as email attachment.

Other practice is related to technical exceptions. Details of technical expcetions are mostly not relevant to business user so detailed exceptions should be emailed to the production support team/developers, while business users may receive email stating that Bot had technical issues.

This requires little bit of extra development for exception handling but it keeps the overall process very streamlined and above all it doesn’t overwhelm the end user with unncessary tehcnical details.

In short you have to use two different set of email distribution lists – business user emails for execution status and reporting and development/support team emails for technical issues.

1. RPA Configuration:

Most RPA tools provide configration features e.g. Environment Variables in Blue Prism. In a shared RPA envinronment many RPA development teams use the environment which could lead to revealing sensitive information like userId, database connection strings, client accounts etc.

Use a custom configuration file i.e. normal text or excel file will avoid anyone from knowing any such senstive details. I prefer using text file since its slightly faster during run-time.

1. Custom logging:

Every Robot MUST have its custom logs created - be it database or normal text file. This simple disciplined practice saves lot of time while troubleshooting during production issue.

Why is this required when tools like Blue Prism log each stage by default?

Generally in production environments the level of logging is set to minimal by the Administrators compared the the level of logging available in development or UAT. Few main reasons for doing this is Performance and Cost. E.g. In production only ‘Errors’ maybe logged by the tool and Information/Warnings maybe turned-off by the RPA Platform Administrator. Hence relying on the default setting of the tool can delay on identifying the root cause in production environment.

Thus custom logging plays important role in resolving issues in produciton environment. However too much logs can make the bot run for longer so ensure that level of custom logging is of the right level e.g. logging only at functional/application boundaries with proper date time stamp

Above guidelines have helped in many production cases and many certified developers or architects who have worked with me have acknowledged their importance while resolving production issues. Agreed that above points result in slightly longer development and testing cycles. But generally 2 to 4 weeks of additional SDLC effort ensures smooth operations in production keeps the end user happy as the cost of support phase is lesser.

One of the key aspects of RPA Automation Solution, like any other software, is Support.

Improper Production Support offering and capability can render the entire Automation Program Strategy ineffective. Defining the Production Support Model is subjective but here will cover few items which can be adapted regardless of which low code automation tool is used.

1. Custom logging:

Every RPA Robot must have its own **custom logs** created. This simple disciplined practice saves lot of time while troubleshooting during production issue and keeps the project sponsor happy as turn around time is very low. Most of the RPA tools like BluePrism UIPath etc, provide detailed logging by defualt by infrastructre admins generally turn them off in production

1. Sending Email Reports and Alerts:

General tendency is to send such reports as **attachments**.

Instead email should have the URL or shared drive location. This has many advantages like data leakage avoidance, performance etc. For technical alerts use another Email DL consisting of the production support team or technical team.

1. RPA Configuration:

Most RPA tools provide configration features e.g. Environment Variables in Blue Prism. In a shared infrastructure this can reveal Backend End-points to other teams which could lead to issues. A custom configuration file i.e. normal text or excel file will easily help in avoiding this.

1. WorkQueues (Blue Prism):

While many RPA robots are implemented using the WorkQueue like constructs but when it comes to production support Bots do need to be designed (State machine) properly which allows resubmission of the failed items enabling operational efficiency in production support efficiency

It is not a complete list by any means but I have worked with many certified RPA developers/architects who have acknowledged importance of these small tick boxes to make the supporting RPA automation a better experience