

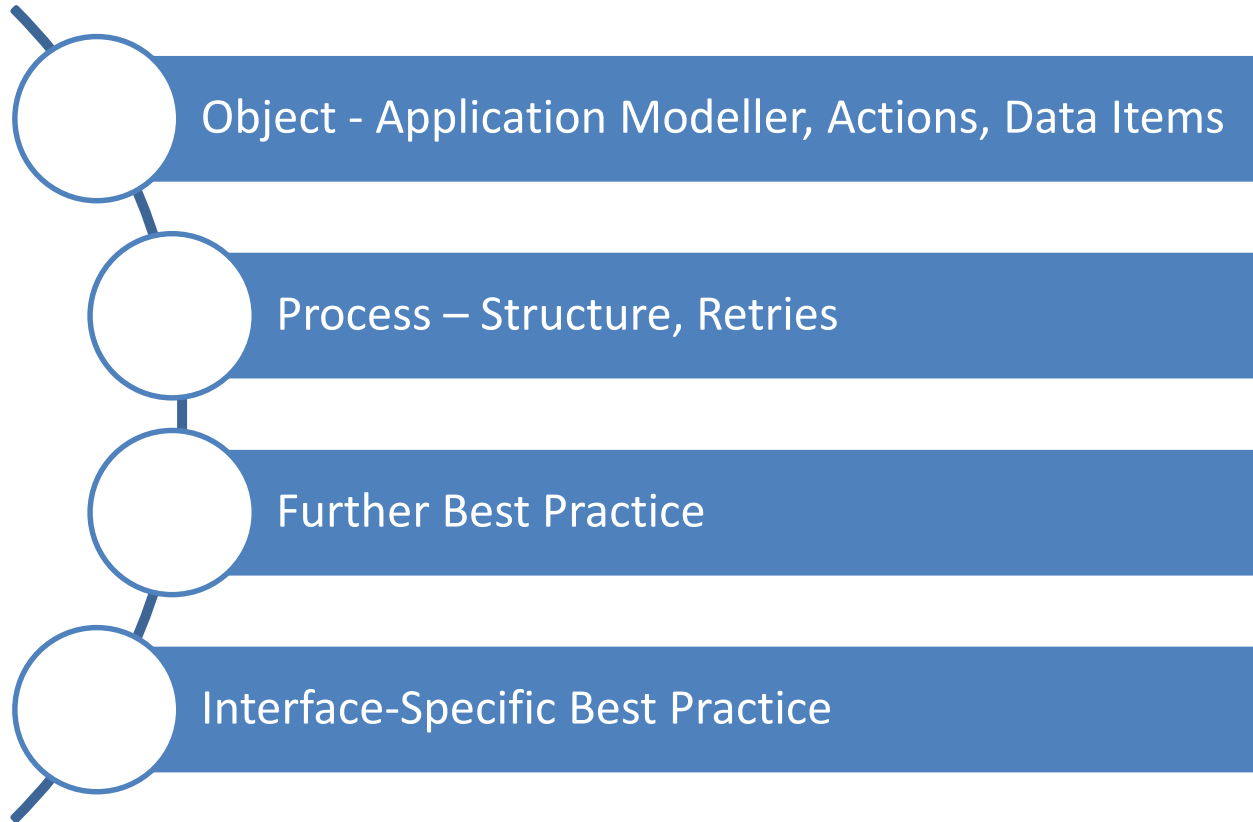
# Delivery Methodology

## Development Best Practice

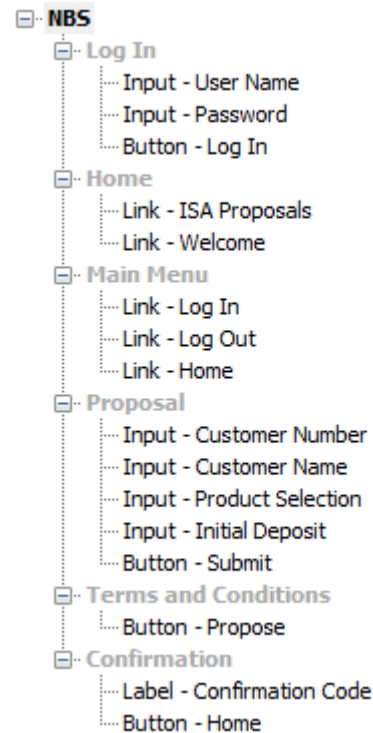
# Development Best Practice

---

## Contents



# Object – Application Modeller



Adheres to local naming convention

Typically this is {element type} – {element name} e.g.  
Button – Submit  
Input – Account Number  
List – Products

Logically Laid Out

Create sections for each part of the screen.  
Makes support easier and mitigates the risk  
of incorrect elements being re-spied

# Object – Application Modeller

Input - Customer Number  
Input - Customer Name  
Input - Product Selection  
Input - Initial Deposit  
Button - Submit  
Forms and Conditions  
Button - Propose

Path	<input checked="" type="checkbox"/>	= (Equal)	/HTML/BODY(1)/
Screen Bounds	<input type="checkbox"/>	:= (Equal)	RECT:625,318,339,8
Tag Name	<input checked="" type="checkbox"/>	= (Equal)	INPUT
Title	<input checked="" type="checkbox"/>	= (Equal)	
Value	<input type="checkbox"/>	= (Equal)	Alastair Bathgate

No customer data

Customer data captured within element attributes could breach data security.

No environment specific data

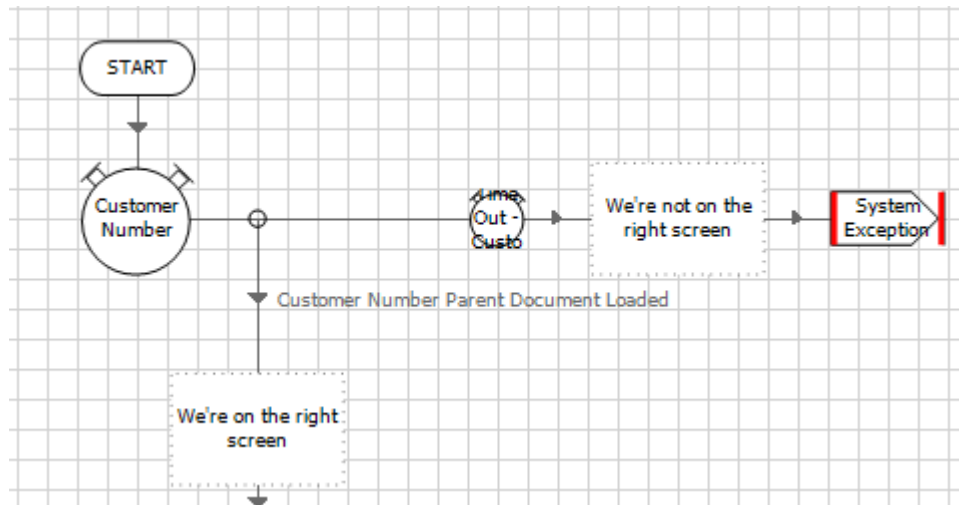
Environment specific data will cause the process to fail when migrated. If required make the value dynamic.

Match Reverse	<input type="checkbox"/>	= (Equal)	True
Parent URL	<input checked="" type="checkbox"/>	= (Equal)	http://UAT.nationalbank.com/proposal.html?ct100%
Path	<input checked="" type="checkbox"/>	= (Equal)	/HTML/BODY(1)/DIV(3)/DIV(1)/TABLE(2)/TBODY(1)/

# Objects - Actions

Wait stage at start of each action

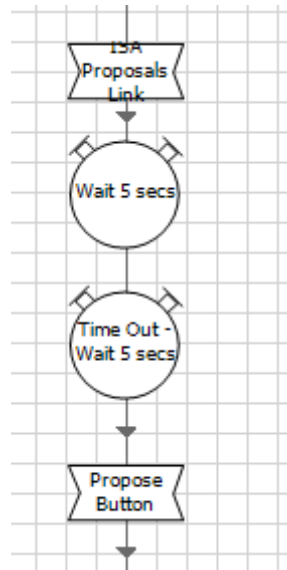
This will confirm the process is on the correct path and absorb system latency to increase the resilience of the process.



Always throw exception on timeout

Do not try and recover the process following the wait stage. Throw the exception and let the process handle it. The process may choose to try again a few times or restart the system or ultimately raise an alert.

# Objects - Actions

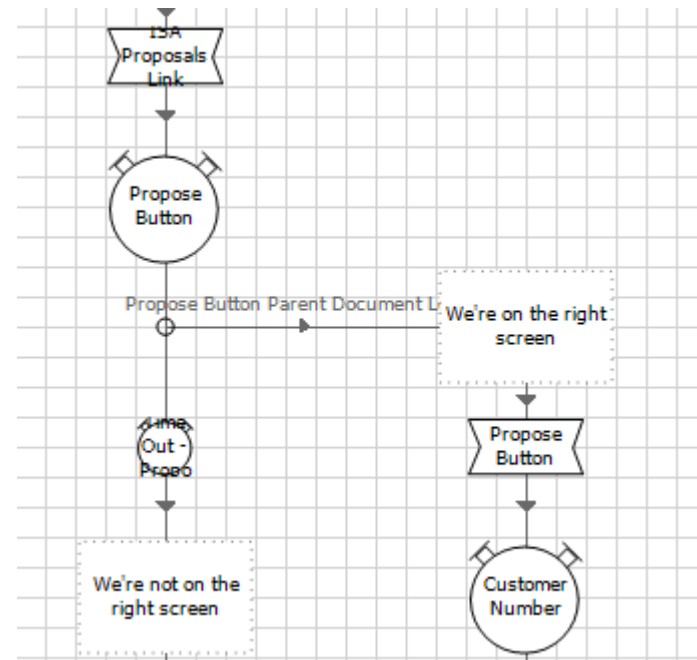


Avoid using arbitrary waits

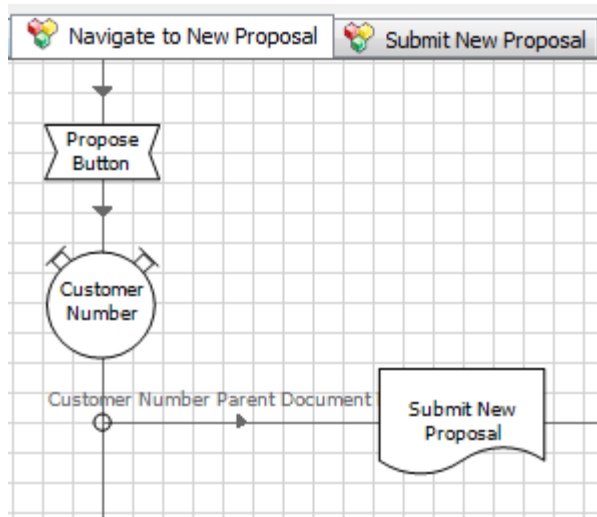
Arbitrary waits should only be used if a screen change cannot be waited for.

Always wait for the screen to change

Use wait stages after Navigate stages or any stage that causes the screen to update. This will absorb any latency but also ensure the process runs at its fastest. In this example there's no point waiting 5 seconds if the system is available after 1.



# Objects - Actions



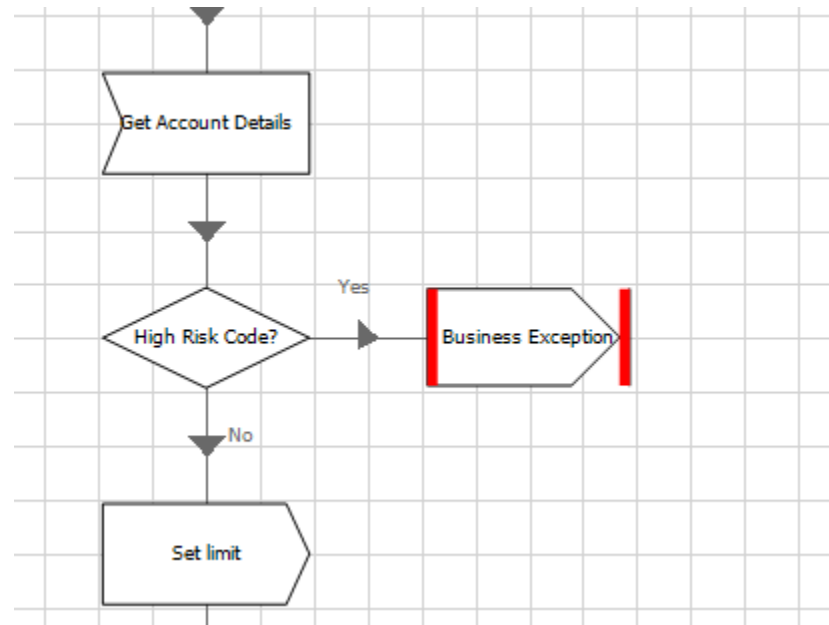
Do not call published actions from within an object

Here the process should call the “Navigate to New Proposal” action and then call the “Submit New Proposal” action.  
This will makes exception handling far easier and actions more reusable.

# Objects - Actions

Do not make business decisions in the object

In this example our action gets account details from the screen and applies a new limit unless the risk codes are of a specific value.

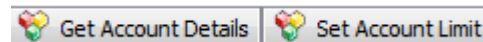


The object design should be such that there are two actions not one.

One action to “Get Account Details” that returns all the account details on the screen and another action to “Apply new limit”.

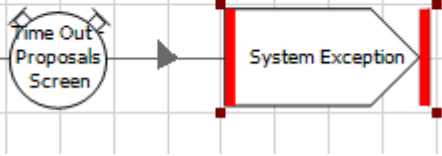
The decision to apply a new limit based on the risk code should be made by the process not the object.

This enables other processes to call the “Get Account Details” action without applying a new limit.





# Objects - Actions



For all exceptions  
provide appropriate  
Type and Detail

The screenshot shows the 'Exception Properties' dialog box in Blue Prism Object Studio. The dialog has a title bar with the Blue Prism logo and the text 'Exception Properties'. The main area contains the following fields and controls:

- Name:** A text box containing 'System Exception'.
- Description:** A large empty text area.
- Exception Type:** A dropdown menu currently showing 'System Exception'.
- Exception Detail:** A text box containing 'Timeout waiting for proposal page to load'.
- ☐ Preserve the type and detail of the current exception
- Help:** A button.
- Stage logging:** A dropdown menu currently showing 'Enabled'.
- OK:** A button.
- Cancel:** A button.

On the right side of the dialog, there is a 'blueprism object studio' logo and an 'Exception Detail' section with the following text:

- Exception detail can be any valid automate expression.
- If you just need some text remember to enclose the text in quote marks.

# Objects - Actions

Provide descriptions to  
Inputs, Outputs and  
Actions

This removes ambiguity and also provides content for auto-generated Business Object Definition (BOD) document.

## 1.8 Submit New Proposal

Completes the new proposal details and presses the submit button

### Preconditions:

On the proposal page

### Endpoint:

On the confirmation page

Parameter	Direction	Data Type	Description
Customer Number	In	Text	Full 8 digit customer number
Customer Lastname	In	Text	Last name of customer. Used to verify customer number
Product Name	In	Text	Product name (ISA, Saving, Current or Loan)
Deposit	In	Number	Deposit amount
Confirmation Code	Out	Text	System generated code provided on successful submission

The screenshot displays the 'Start Properties' and 'Page Information' windows in the Blue Prism Object Studio. The 'Start Properties' window shows the Name as 'Start' and a blank Description field. The 'Page Information' window shows the Name as 'Submit New Proposal' and the Description as 'Completes the new proposal details and presses the submit button'. It also lists Preconditions as 'On the proposal page' and Postconditions as 'On the confirmation page'. At the bottom, there is a checkbox labeled 'Publish this page as an Action' which is checked, and buttons for 'Help', 'OK', and 'Cancel'.

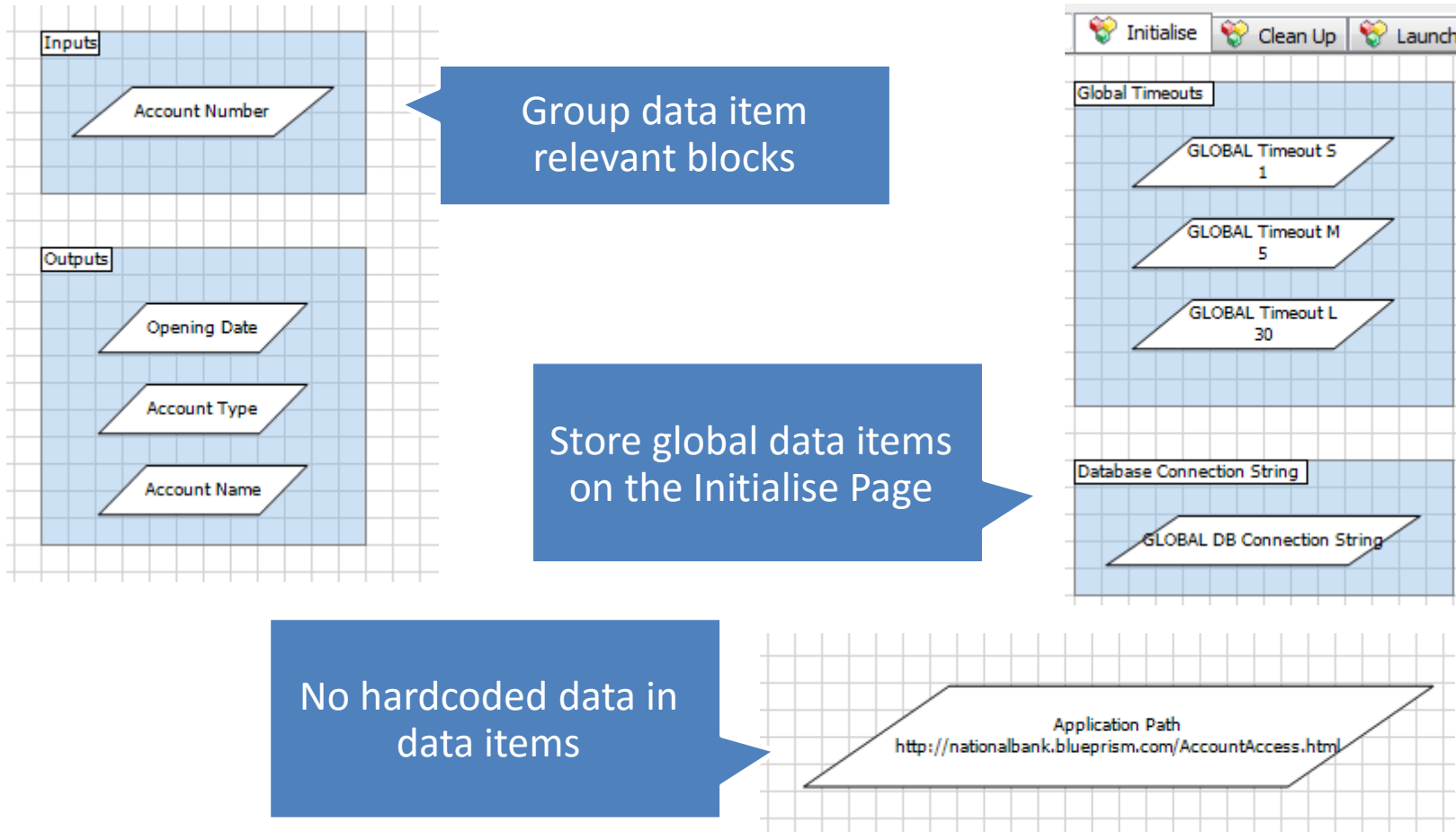
Name	Description
Customer Number	Full 8 digit customer number
Customer Lastname	Last name of customer. Used to verify customer number
Product Name	Product name (ISA, Saving, Current or Loan)
Deposit	Deposit amount

Preconditions  
On the proposal page

Postconditions  
On the confirmation page

Help ☒ Publish this page as an Action OK Cancel

# Objects – Data Items



Use input parameters and have the process provide the values. If need be, the process can consume an environment variable for parameters that may change.

# Objects - Exposure

---

Run Mode

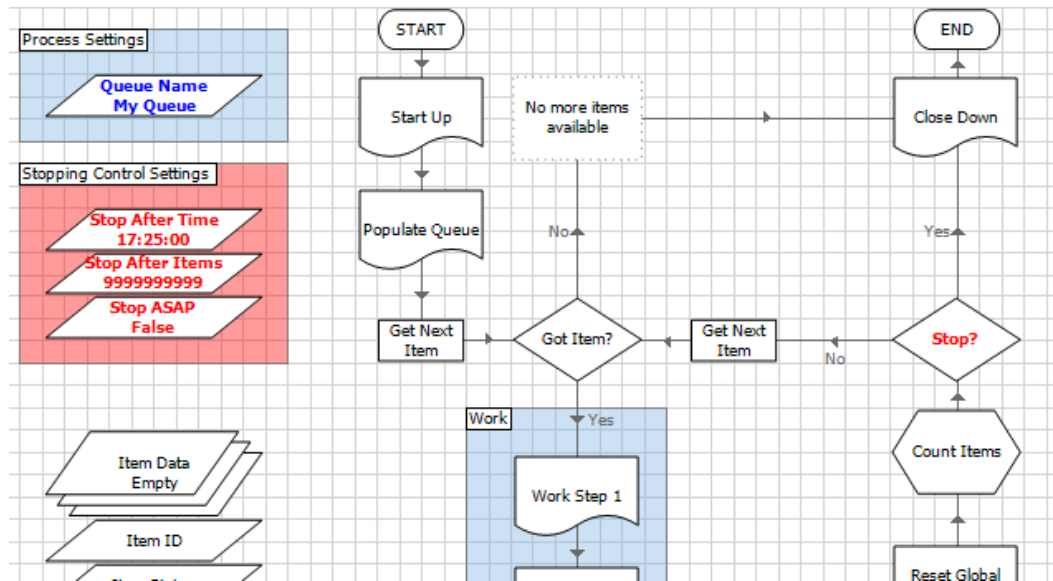
- ☐ Foreground If this object should never have more than one instance running at the same time on the same resource, set to Foreground. This business object can run at the same time as other background business objects.
- ☐ Background If this object is designed to support multiple instances running at the same time on the same resource, set to Background. This business object can run at the same time as a foreground business object or other background business objects.
- ☒ Exclusive If this object should never be run at the same time as any other object, and requires exclusive access to the desktop, set to Exclusive. This business object can not run at the same time as any other business object.

Always set an objects  
exposure

# Process - Structure

Use standard Blue Prism templates or templates provided by the local design authority

The standard logic of the templates enables familiarity that makes support easier.



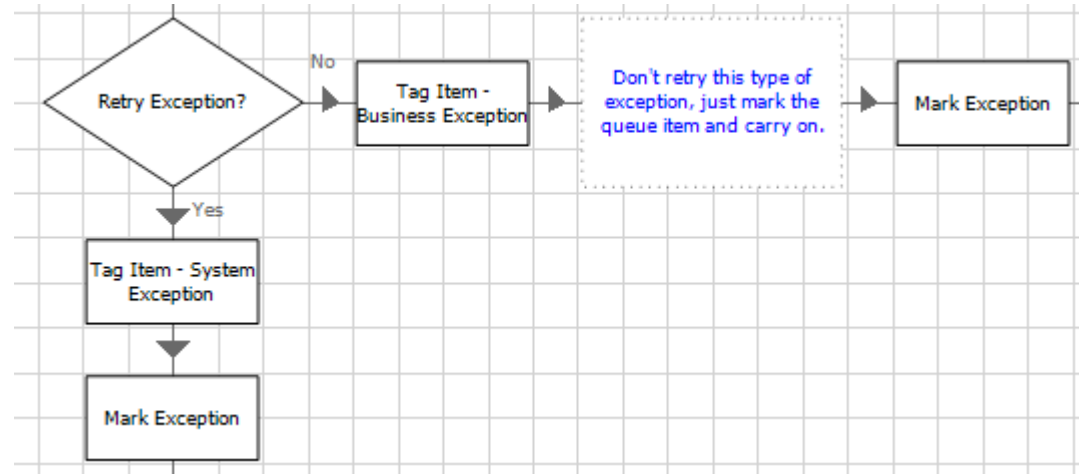
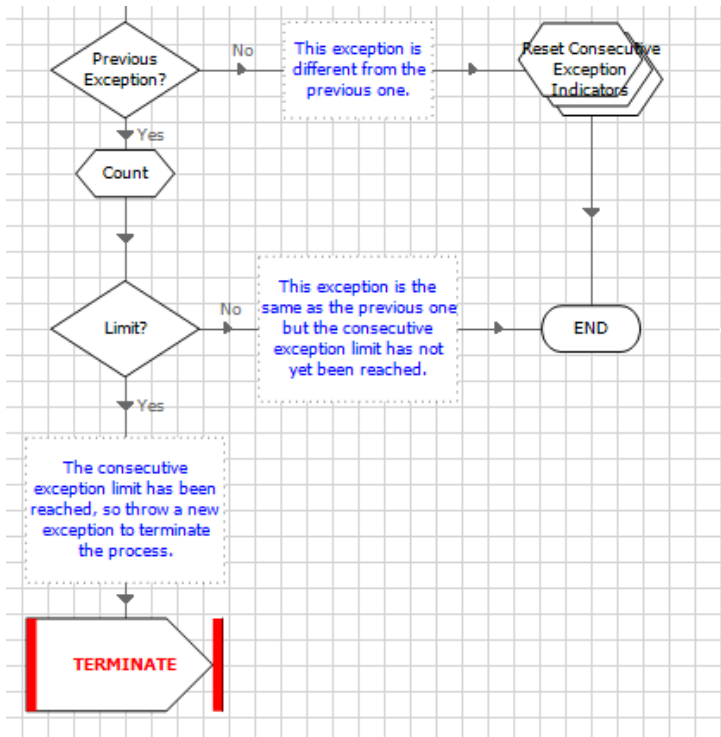
Main page should contain high level process steps

All process detail and decision should be divided into logical sub pages

# Process- Retries

## Correct use of retries

If the work queue permits retry system exceptions. Do not retry business exceptions (see Basic Template)



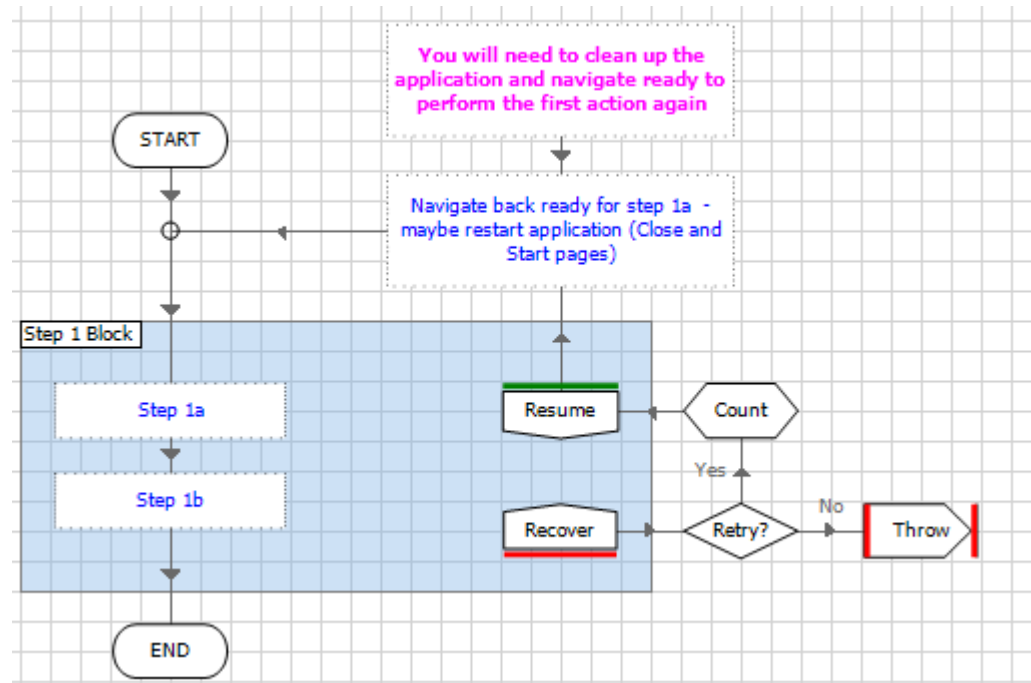
## Check for consecutive system exceptions

The same system exception across consecutive cases could be a sign that the system has changed. (see Basic Template)

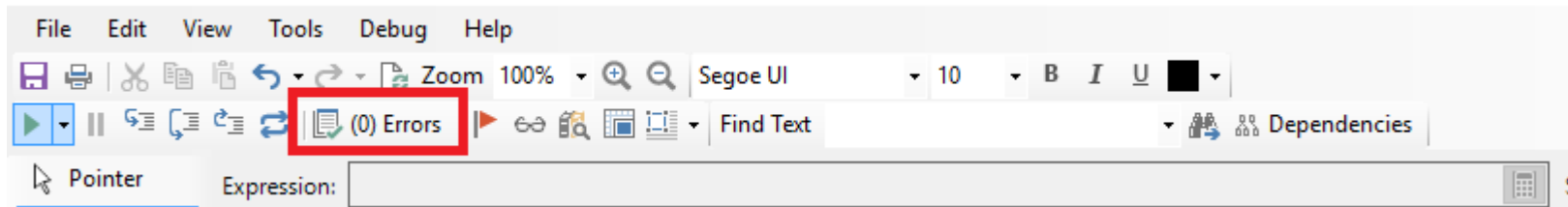
# Process - Retries

## Correct use of retries

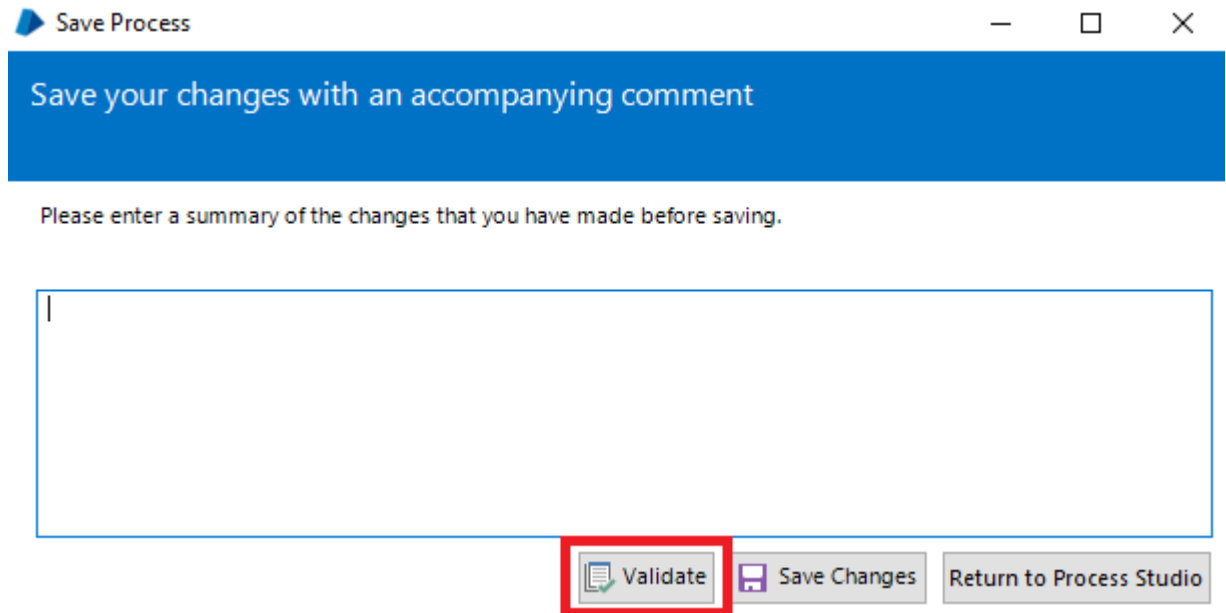
Where possible retry system exception within the process. This may require special navigation or even a restart of the system



# Further Best Practice



Always check for errors  
or validate when saving.

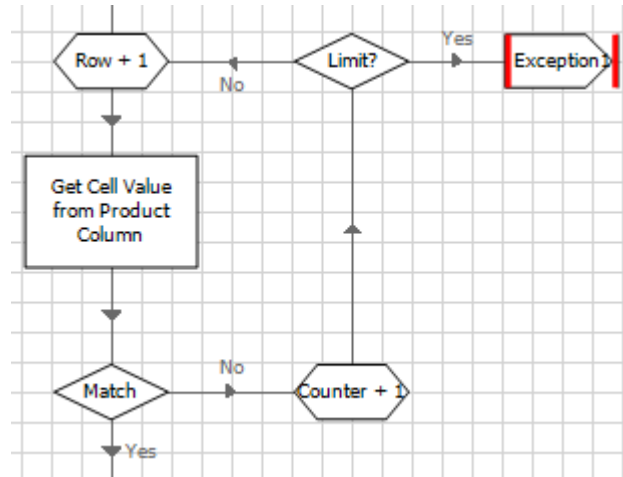




# Further Best Practice

## Manufactured Loops have defence mechanism

When creating your own loops always create a counter and throw an exception if an excessive limit is reached. This will prevent the flow ever entering an infinite loop.



Stage logging: ☒ Enabled ☐ Don't log parameters on this stage

Stage logging adheres to local design control

## Avoid use of drive letters when referencing folders

Data Type	<input type="text" value="Text"/>
Initial Value	<input type="text" value="\\\\SERVERNAME\\shraedfolder\\spreadsheet.xlsx"/>
Exposure	<input type="text" value="Environment - Read the corresponding Environment Variable from System Manager"/>

# Interface-Specific Best Practice

---



Java Automation Guide



Browser Automation Guide



Mainframe Guide