



Prodapt powering
global telecom

Automation of RPA Code Review Process

Accelerate RPA Development Lifecycle

Credits

Srinath R

Murugavel D

Sarvagya Nayak

Manual code review of bots makes the RPA development lifecycle tedious, time consuming and erroneous

For a RPA CoE, it is very important to review the bot code before it moves into production in order to:

Minimize post shipment defects

Ensure standardization in all the bots deployed

The traditional way of reviewing RPA code includes:

1. Manual line-by-line code review
2. Analyzing each and every property
3. Examining the entire list of variables & arguments
4. Reporting all deviations

But the usual manual code review is a very daunting task. A typical bot code is very complex consisting of multiple workflows and each workflow, in turn, consisting of:

Hundreds of - variables, arguments, activities and message boxes

Logic for exception handling, custom logging, queues and credential management

Manual code review of bot has its own challenges like:

- **Erroneous** - Automations developed by business users are highly error-prone and often does not comply to standards
- **Time consuming** - Reviewing a code requires validating each element line-by-line and therefore is very daunting and takes few hours to few days
- **Late identification of missing validations** - Missing some of the validations during code review leads to defects being identified in UAT and some of them are captured once the bot is in production
- **Hot fixes** - Post implementation defects and non-compliance to standards lead to root cause analysis and hot fixes during the stabilization period

Even though code review is a highly critical step, it is crippled with challenges with its traditional approach of being manual. It is recommended to automate this step for enhancing overall efficiency.

Code reviewer bot – most efficient tool to address code review challenges in bot development

XML file

RPA bot code extracted as a XML file

- RPA code can be opened using corresponding RPA tool only
- It is recommended to extract bot code as XML file

Code Reviewer Bot



- Develop business rules to be validated by bot
- Implement 'code reviewer bot', which will validate as per these rules
- Code reviewer bot should be capable of validating each activity/line of the RPA project and reading any number of workflows

Validation report

- Comprehensive report with summary of all the finding and missing standards.

Key capabilities to be built in code reviewer bot

- Should be a reusable component with functionality accessible across business units and processes
- Should be a standalone bot without dependency
- Able to work on bot code of any RPA tool i.e. tool agnostic
- Easy to use - Should be usable by RPA developers with minimal experience
- Should ensure adherence to organization's bot development standards
- Ability to run in background

Snapshot of workflow extracted as XML

```
<Variable x:typeArguments="scg:IEnumerable`1" Name="displayname" />
<Variable x:typeArguments="x:String" Name="actName" />
<Variable x:typeArguments="x:String" Name="varAction" />
<Variable x:typeArguments="x:String" Name="argAction" />
<Variable x:typeArguments="xl:GenericValue" Name="killAction" />
<Variable x:typeArguments="x:String" Name="argumentName" />
</Sequence.Variables>
<ui:ReadTextFile Content="[xmlInputData]" DisplayName="Read text file" FileName="[in_xmlInputPath]"
sap2010:WorkflowViewState.IdRef="ReadTextFile_1" />
<Assign sap2010:WorkflowViewState.IdRef="Assign_31">
  <Assign.To>
    <OutArgument x:typeArguments="x:String">[filename]</OutArgument>
  </Assign.To>
  <Assign.Value>
    <InArgument x:typeArguments="x:String">[Path.GetFileName(in_xmlInputPath)]</InArgument>
  </Assign.Value>
</Assign>
<If Condition="([filename.Length>30]" DisplayName="Check Length of the filename" sap2010:WorkflowViewState.IdRef="If_21">
  <If.Then>
    <Assign sap2010:WorkflowViewState.IdRef="Assign_49">
      <Assign.To>
        <OutArgument x:typeArguments="x:String">[filename]</OutArgument>
      </Assign.To>
      <Assign.Value>
        <InArgument x:typeArguments="x:String">[filename.Substring(0,30)]</InArgument>
      </Assign.Value>
    </Assign>
  </If.Then>
</If>
<Assign sap2010:WorkflowViewState.IdRef="Assign_1">
  <Assign.To>
    <OutArgument x:typeArguments="s:String[]">[xmlLineArray]</OutArgument>
  </Assign.To>
  <Assign.Value>
```


Code reviewer bot categorizes and validates the data analyzed in the XML file

1 2 3 4

2. Categorize Data

Categorize the details extracted from analyze phase to group them into:

- The information to be sent for validation logics
- Dissect the information needed for both detailed and summary report
- Classify to group the extracted information to data items, activities, standards and properties

3. Validate Data

Validation includes various parameters:

- Name, type and naming convention of variables and arguments
- Usage of comments
- List of all activities, log message and logging levels
- Usage of configuration files
- Validation of hardcoded passwords and parameters
- Usage of custom log fields

This sample of an XML file contains 4 categories in which the data can be **categorized** – arguments, annotations, logging and exception handling. The data within these categories are further **validated** as per the standards specified by the CoE and the business rules.

```
<x:Members>
  <x:Property Name="inStreetAddress" Type="InArgument(x:String) />
  <x:Property Name="inCity" Type="InArgument(x:String) />
  <x:Property Name="inState" Type="InArgument(x:String) />
  <x:Property Name="inZipCode" Type="InArgument(x:String) />
  <x:Property Name="USPCflag" Type="OutArgument(x:Boolean) />
</x:Members>

</TextExpression.ReferencesForImplementation>
<Sequence sap2010:Annotation.AnnotationText "This workflow outputs a settings
Dictionary with key/value pairs to be used in the project. Settings are read from
local config file then fetched from Orchestrator assets. Assets will overwrite
config file settings" DisplayName="Get All Settings" sap2010:WorkflowViewState
.IdRef="Sequence_2">
  <ui:LogMessage DisplayName="Log message" sap2010:WorkflowViewState.IdRef
="LogMessage_4" Level="Info" Message="Starting" />
  <Assign DisplayName="New dictionary" sap2010:WorkflowViewState.IdRef
="Assign_1">
    </TryCatch.Try>
  </TryCatch.Catches>
  <ui:Message DisplayName="Log message" sap2010:Workflow
ViewState.IdRef="LogMessage_1" Level="Trace" Message="No assets defined for the
process" />
</ActivityAction>
</Catch>
</TryCatch.Catches>
</TryCatch>

```

1. Argument name
2. Argument direction
3. Argument type
4. Annotation
5. Logging message
6. Logging level
7. Exception handling

Code reviewer bot transforms the validation efforts into detailed reports for actionable insights

1 2 3 4

4. Transform Data

The code reviewer bot should generate 2 detailed reports

Detailed Report

Consists of separate sheets for each workflow/script with a detailed list of variables, arguments, activities, delays, hardcoded passwords, usage of queues, exception handling and messages logged to visualization databases along with their logging levels.

Summary Report

An easy to understand summary report categorizing each deviation and their severity.

Variable Name	Variable Type	Match/Mismatch	Default	Activities	Display
boolFirstExec	Boolean	Match		Log message	
IstTransactionItems	List(scg:Dictionary(x:String, x:O	MisMatch		Open browser	
exBusinessRuleError	BusinessRuleException	Match		Do	
iTransactionItemsProcessed	Int32	MisMatch			
ShouldStop	Boolean	MisMatch			
URI	String	MisMatch			
Test	String	MisMatch			

Detailed Report

A mismatch here indicates:

Either the naming convention for "Test" is not followed **or**

Variable type for "Test" is "String" whereas it should be something else

S.No	Item	Status	Severity
1	Total Number of Workflow Files	12	N/A
2	Total Number of Variables	98	N/A
3	Total Number of Arguments	43	N/A
4	Total Number of Activities	731	N/A
5	Does the workflows follow variable naming conventions?	Red	Medium
6	Does the workflows follow argument naming conventions?	Red	Medium
7	Does the workflows contain manual delays?	Red	Medium
8	Does the workflows contain Message Box?	Green	High
9	Does the workflows contain write line?	Red	Medium
10	Does the workflow uses queues concept?	Green	Medium
11	Has the should stop activity been used ?	Green	Medium
12	Has all the mandatory log fields included?	Green	High
13	Does the workflows have hard-coded URLs?	Green	Medium
14	Does the workflow has hard-coded passwords?	Red	High
15	Does the workflow has commented code?	Red	Low
16	Does the workflow contain input dialog box?	Green	High
17	Does the workflow contain kill process activity?	Red	High

Summary Report

- 1 Red means "variable naming conventions" not followed
- 2 This indicates "including mandatory log fields" is of high importance for this code
- 3 Since for a high severity item status is red, this item needs immediate action

Benefits of using bot in code review process

🎯 30% - Increase in deviation identification efficiency

🎯 ~99% - Time saved in automated code review process as compared to manual one

	Small process	Medium process	Complex process
 Manual code review	3-4 hrs	5-6 hrs	6-8 hrs
 Bot enabled code review	~1.5 mins	~2.5 mins	~5 mins

🎯 Ensures 100% compliance to standards

🎯 Improved code maintainability

Get in touch

USA

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

Dallas: 1333, Corporate Dr., Suite 101, Irving
Phone: +1 972 201 9009

New York: 1 Bridge Street, Irvington
Phone: +1 646 403 8161

CANADA

Prodapt Canada Inc.
Vancouver: 777, Hornby Street,
Suite 600, BC V6Z 1S4
Phone: +1 503 210 0107

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
München: Brienner Straße, 80333
Phone: +31 (0) 70 4140722

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

IIT Madras Research Park II,
3rd floor, Kanagam Road, Taramani

Bangalore: “CareerNet Campus”
2nd floor, No. 53, Devarabisana Halli,
Phone: +91 80 4655 7008

THANK YOU!