Automation overview: (time to dev, test, etc)

Robots type: Unattended

Level of human intervention required: none

Use of Orchestrator: for queue and for manage re-run transactions

Exceptions recorded in automation process: almost any kind include lost connection to production server / login/timeout … etc

Errors identified in the automation process:

Challenges identified in the automation process:

Lessons Learned: Any adjustments made to facilitate the automation process and any steps taken to shift from the human way of working to the automatic one. Any activity performed to improve the

As Is process and to enable higher rates of automation of the process: ➢ Process Assumption ➢ Input data assumption ➢ Number or types of input to be received ➢ Skipping the login interface and collecting backend details ➢ Extracting backend data without opening the file ➢ Data conversion / formatting

Reporting: The details and format of the logging mechanism available in the workflow have to be specified here, whether it is a local log report or the Orchestrator log). The format should be specified by the business users.

Workflow and scripts: A brief overview of each workflow and the sequence in which it is executed should be provided here.

ExceptionHandlerImpl.ExceptionHandlerArgsImpl

{

ActivityInfo=[Name=Read range settings,

ActivityId = 1.49,

ActivityInstanceId = 6,

TypeName=UiPath.Excel.Activities.ReadRange],

Arguments=ReadOnlyDictionaryInternal<string, object>

{

{

"Workbook",

null

},

{

"DataTable",

null

},

{

"Range",

null

},

{

"Password",

null

},

{

"SheetName",

"Settings"

},

{

"WorkbookPath",

"Data\\Config.xlsx"

}

},

Exception=[System.IO.IOException: The process cannot access the file 'C:\Users\Alex\Documents\UiPath\Advanced Traning\Generate Yearly Report\Data\Config.xlsx' because it is being used by another process.

at System.IO.\_\_Error.WinIOError(Int32 errorCode,

String maybeFullPath)

at System.IO.FileStream.Init(String path,

FileMode mode,

FileAccess access,

Int32 rights,

Boolean useRights,

FileShare share,

Int32 bufferSize,

FileOptions options,

SECURITY\_ATTRIBUTES secAttrs,

String msgPath,

Boolean bFromProxy,

Boolean useLongPath,

Boolean checkHost)

at System.IO.FileStream..ctor(String path,

FileMode mode,

FileAccess access,

FileShare share,

Int32 bufferSize,

FileOptions options,

String msgPath,

Boolean bFromProxy)

at System.IO.FileStream..ctor(String path,

FileMode mode,

FileAccess access,

FileShare share,

Int32 bufferSize,

Boolean useAsync)

at MS.Internal.IO.Zip.ZipArchive.OpenOnFile(String path,

FileMode mode,

FileAccess access,

FileShare share,

Boolean streaming)

at System.IO.Packaging.ZipPackage..ctor(String path,

FileMode mode,

FileAccess access,

FileShare share,

Boolean streaming)

at System.IO.Packaging.Package.Open(String path,

FileMode packageMode,

FileAccess packageAccess,

FileShare packageShare,

Boolean streaming)

at DocumentFormat.OpenXml.Packaging.OpenXmlPackage.OpenCore(String path,

Boolean readWriteMode)

at DocumentFormat.OpenXml.Packaging.SpreadsheetDocument.Open(String path,

Boolean isEditable,

OpenSettings openSettings)

at ClosedXML.Excel.XLWorkbook.LoadSheets(String fileName) in C:\Projects\ClosedXML\ClosedXML\Excel\XLWorkbook\_Load.cs:line 44

at UiPath.Excel.WorkbookFile..ctor(String workbookPath,

String password,

Boolean createNew)

at UiPath.Excel.Activities.WorkbookActivity`1.ConstructWorkbook(String path,

String password,

Boolean createNew)

at UiPath.Excel.Activities.WorkbookActivity`1.BeginExecute(AsyncCodeActivityContext context,

AsyncCallback callback,

Object state)

at System.Activities.AsyncCodeActivity.InternalExecute(ActivityInstance instance,

ActivityExecutor executor,

BookmarkManager bookmarkManager)

at System.Activities.ActivityInstance.Execute(ActivityExecutor executor,

BookmarkManager bookmarkManager)

at System.Activities.Runtime.ActivityExecutor.ExecuteActivityWorkItem.ExecuteBody(ActivityExecutor executor,

BookmarkManager bookmarkManager,

Location resultLocation)],

RetryCount=0,

Variables=ReadOnlyDictionaryInternal<string, object>

{

}

}