How to Approach and Start a Document Understanding Project

Introduction

Every organization uses different documents daily to perform its critical business activities. Processing documents can be time-consuming and requires much effort. Document Understanding solutions help reduce manual effort and time by processing documents with higher accuracy and reliability. However, every document understanding solution comes with its own set of challenges. Some of the main challenges in document understanding projects are the variations of documents, complex business rules, and the technology required to process the high volume of documents. It is essential to understand these challenges at the initial stage of the project to build a cost-effective, reliable, efficient, and scalable automation solution.

Requirements

Before getting started, you'll need the following

- Basic understanding of the UiPath Document Understanding Framework (https://docs.uipath.com/document-understanding/docs/introduction)

Building the Scenario

The complexity of a document understanding project can range from a simple, straightforward process to a highly complex project with multiple document types in different languages. Identifying all the possible scenarios at the initial stage of the project help RPA Solution Architects and developers plan the solution and think of the technology needed to process the documents. At a high level, a few factors require attention in any document understanding project. These can be considered a checklist to gather requirements for any document understanding project.

Category	*Description*	*Comments*
Source	What is the source of the documents?	SharePoint, One Drive, Shared drives
	How to identify the documents that require processing?	
	What is the approach taken to store the documents?	Folder structures, file naming patterns, Folder naming patterns
	What is the average number of documents processed in each iteration?	
	What is the frequency of processing the documents?	Hourly/ Daily/ Weekly
File Structure	What are the different file types	PDF, Image files, Excel, Word
	The average number of pages in a document	
	Can the files be password protected	
	How is the data presented in these files?	Computer-generated files, scanned documents, photos, handwritten text, signatures
Classification	What are the different document types	Invoices, Purchase Orders, etc.
	Can a single file contain multiple document types?	
	Can multiple document types fall under a specific classification?	For example, the classification might be "Legal Documents." But are there legal emails, contracts, agreements, and court orders that fall under this classification type
	Is it required to split the document (files containing multiple types)?	
Document Type Structure	What is the layout of each document classification identified	Structured, semi-structured, unstructured
	Can documents with multiple layout structures fall under a given classification type?	For example, a classification type that says "Application Forms" may sound like it contains only structured documents. However, does it also get unstructured documents, like emails?
	How many different layouts available for each document type	E.g., If the document is an invoice, how many vendors submit invoices of different layouts?

Required Data	What information should be extracted from the documents?	
	What are the different ways of representing the data?	Structured forms, semi-structured, unstructured, text patterns, handwritten, signatures, barcodes
Validations	What are the business rules to verify the accuracy of the data presented in the document?	Use this information in building your validation logic to verify the extracted data
	What data transformations are required?	
	Is manual verification possible for data extracted with low confidence?	
Post-processing	What is the destination system	
	What are the SLA levels in updating the destination systems?	
	What information is required for updating the systems?	

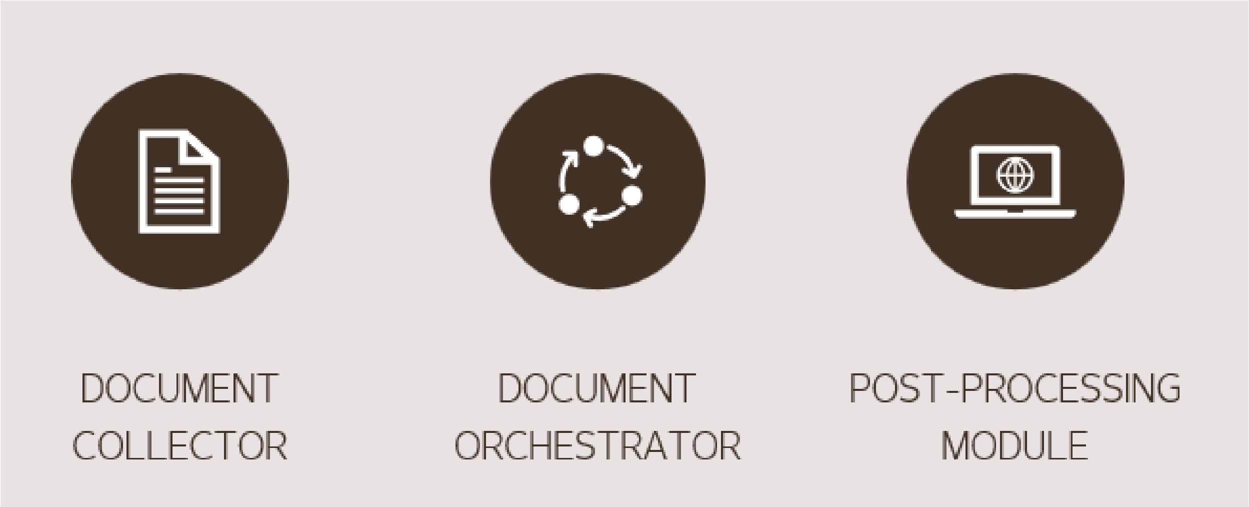
Let's now look at how we can use this checklist to gather information and plan the way forward.

Step 1: Requirement Gathering

Like any other RPA project, understanding the clear requirement is vital for document understanding projects. Using the checklist provided above gives all the insight we need about the documents involved in the process. Talk to business users and understand how they process each document type, the information extracted, and how they validate it before updating downstream applications. Other than capturing the business requirement, requesting sample documents of each identified document type is also important. Request for all possible variants to uncover the points mentioned in the "File Type Structure," "Classification," and "Data Extraction" sections in the checklist. In addition, going through the files most of the time uncovers more hidden scenarios that require addressing during development.

Step 2: Planning the Solution

Compare the samples received with the business rules described to identify the techniques required to process the document. The checklist provides excellent insight into some of the techniques needed. A document understanding solution typically needs to have three main modules.



Document-processing-modules

As a best practice, the three-module architecture should remain the same across all the Document Understanding projects. The architecture of each module may depend on the information identified through the checklist mentioned above. The data handover between the modules requires a data store such as the Orchestrator Queue. The use of Queues for the data transition has more benefits than other approaches because of the built-in retry capabilities and the ability to run multiple jobs on the same queue without duplication.

This article will focus more on the Document Collector and the Document Orchestrator modules.

Step 2.1: Structuring the Document Collector

The section "Source" in the checklist provides all the information on how the files are stored, access methods, and the required logic to identify new files. It is crucial to plan the architecture of the Document Collector module based on the information gathered from this section. The checklist provides information on how to identify new documents. However, we need to have a mechanism to ensure we don't process the same document multiple times in case the process runs into an unexpected error and needs restarting. The source system may only provide us with information to identify new documents. Hence, it is essential to use a secondary data store to maintain a record of items that got pushed into the Document Orchestrator module. The monitoring can be done through Orchestrator Queues, Data Services, Excel, or a SQL database.

Step 2.2: Structuring the Document Orchestrator

The Document Orchestrator module consists of several sub-sections that require configuration during development.

- Defining the document types and the data fields required for each type
- Digitization of the documents
- Classification and document splitting strategy
- Data extraction strategy
- Model retraining strategy (if applicable)

The requirement for these subsections must be clear before proceeding on any Document Understanding project.

Defining document types and data fields

Defining the document types and the fields depend on the insights gathered through the "File Structure," "Classification," "Document Type Structure," and "Extract Data" categories. It is essential to define the document types at the most granular level according to business requirements and group them accordingly. One of the other important points is the types of documents that fall into one specified category. For example, in Invoice processing, the type "Invoice" always contains only invoices. However, we come across scenarios where each defined type may contain multiple types of documents. For example, "Requisition for Release of Information" may contain court orders, court reports, email communications, requisition forms, and many more. Identifying these scenarios is essential when defining the document types and planning the classification.

Digitizing Documents and Use of OCR Engines

Document digitization requires an OCR engine to convert scanned documents or images to machine-readable format. The selection of the OCR engine depends on how accurately it extracts data from the documents. One OCR may work well on one project, but it may not work well on other types of documents. Hence, identify OCR engines that support all identified document variations (computer-generated, handwritten text, scanned documents, and photos). Next, perform an OCR benchmarking test to compare multiple OCR engines on actual samples to identify which OCR engine performs the best. The advantage of benchmarking test is that it also helps identify which scenarios and data fields require more focus in validating the extracted data.

Classification and Requirement for pre-Trained Classification Models

The use of the classification model depends on the nature of the documents. Following is a simple guide to performing the selection.

- Use Keyword Based Classifier if the documents are straightforward and if you can easily define the unique keywords
- Use Intelligent Keyword Classifier if the file contains multiple document types that require splitting
- Use Machine Learning Classifier if the documents are mainly unstructured and have many variations making it challenging to specify unique keywords

However, depending on the use case, you may also find scenarios that require to perform splitting on largely unstructured documents with many variations. In such cases, approach the classification in a way where you use multiple classifiers to support all the requirements.

Identify the primary classifier based on the required functionality and accuracy, and use other classifiers to support the primary classifier to generate the expected outcome

Data Extraction and selection of Extraction Models

The selection of the extraction method depends on several factors described in the "Data Extraction" section on the checklist. The extraction method mainly depends on the methods used to represent data and the document layout. Use the following questions as a guide to decide which extractors to use.

- What is the layout of the document?
  - Structured and has a fixed number of rows in tables – Form Extractor Template or Forms AI
  - Structured, but does not have a fixed number of rows in tables – UiPath Form Extractor (multiple templates), Forms AI, Form Extractor along with Regex Extractor
  - Semi-structured and many layouts – ML Extractor (custom or out-of-the-box)
  - Unstructured – language analysis models based on the values to be extracted (NER, Semantic Analysis, Classification)
- Is it required to check the availability of signatures and checkboxes?
  - Use Form Extractor
- Is it required to identify a signature and extract it for comparison?

    - Object detection models or Form Extractor (crop and extract using signature area coordinates)
- Is there any pattern the bot can follow to identify elements to extract?
    - Regex Extractor
- Can we use multiple extractors?
    - If you see a combination of all the above, we can use multiple extractors as required
    - We can use multiple extractors in scenarios where one extractor is not performing well on certain documents/ patterns. The secondary extractor can assist on such occasions.

Model Training and Retraining

Training a model for classification (Keyword, Intelligent Keyword, or ML Classifier) is performed based on the documents and the task we need to achieve. Consider the following when performing the initial training of the classifiers.

- For large documents (a file that belongs to one type but has several pages)
    - Use Keyword Classifier and manually define the unique keywords
- For large documents (a file that contains multiple document types and has several pages that belong to each type)
    - Split the file into multiple documents that contain only a single page
    - Use those single-page documents to train the classifier for each type. (This way, the classifier learns to understand each page individually and predict accurately)

UiPath also provides an extensive collection of out-of-the-box models for data extraction. It is essential to evaluate whether the available models can serve the requirement when there is a need for machine learning extractors. Consider the following when performing the initial training on the out-of-the-box models.

- Collect a good number of documents for each document type you wish to train. The higher the number of samples, the better the model can predict.
- Collect at least 10-50 documents for each layout for training documents with different layouts (e.g., Invoices)

Conclusion

Large amounts of data in organizations are stored in documents for various reasons. The document processing projects may contain complex documents and business logic that require much configuration. However, following a checklist to capture all those scenarios help ease the development activities on document understanding projects. Further, having the correct information also helps to plan the solution to use the required technology and techniques to ensure the delivery of a scalable and reliable automation solution.