

# Best Practices for PDD Creation

## General:

- When creating a UiPath Process Design Document (PDD), begin by thoroughly understanding the process.
- Clearly define inputs, outputs, and business rules.
- Break down the process into manageable steps, use diagrams to illustrate workflows, and provide detailed descriptions.
- Include error-handling mechanisms and document any exceptions.
- Regularly update the PDD to reflect changes.
- Collaborate with stakeholders to ensure accuracy and completeness.
- Include version control to track document changes over time.
- Document dependencies, such as external systems or APIs, and consider scalability for future enhancements.
- Keep it concise and focused on essential information to avoid unnecessary complexity.
- Detail the data flow, specifying how information moves through the process.
- Use a standardized naming convention for clarity.
- Regularly review and update the PDD as the project progresses, ensuring it remains a reliable reference for both developers and stakeholders.
- Foster collaboration by involving developers in the PDD creation process to harness their insights.
- Lastly, maintain a balance between detail and conciseness for easy readability, avoiding unnecessary information overload.

## TO - be process description

- Make sure the scope of the project is clearly defined and documented and, if it's not, raise this issue in the first meetings with the stakeholders
- Map the system/app requirements.
  - Name and version of the application required to run the process.
  - Tech/Dependencies - Does the application make use of any plugins which will also need to be installed?
  - Environment - The operating system or virtualization software (e.g. Citrix) that the application will need to run in
  - Authentication - Does the application require credentials or an SSO login.
  - Bot role - What is the minimum level of access the Bot will need e.g. read-only or write.
- Document bot schedule
  - Start time.
  - Time zone
  - Repetition - How often should the Bot be run e.g. every 1 hour.
  - Repetition end time - At what time should the Bot stop running.
  - Frequency - How often should this schedule be run
- Document/provide examples of inputs/outputs (datasets from Excel/CSV files, config files, pdf/image files, emails, etc.)
- The to-be process diagram should depict the interaction between components (package / robots, Orchestrator queues, and running order)
- The list of to-be process steps should represent a task bot in the overall process. A step should contain:

- Step number - e.g. "Step #1"
- Name of the task - e.g. "DownloadPricesBot"
- The path to the task bot in the Control Room - e.g. "BotsFinanceS&P500"
- Description of the task bot - e.g. "This is the main task bot which orchestrates the overall logic of the solution."
- Detailed steps that the task bot should carry out - For example:
  - Call "01\_Read\_Config" subtask to load the config variables into a dictionary.
  - Call "01\_Setup\_BotRunner" subtask to set the resolution and clear the desktop of the available Bot Runner
  - Open Yahoo Finance page for S&P 500 historical prices as per the "sp500\_url" config variable
  - Create a new Excel file from the "sp500\_output\_dir" config variable.
  - Give new Excel file a filename in the format "yyyy-mm-dd\_hhmm - prices\_download"
  - Loop through each of the paginations when on the Yahoo Finance page:
  - Extract the table data into the new Excel File
  - Making sure to append the data.
  - Save the Excel file.
  - Call "06\_Send\_Email" to send a completion email to the business team.
- Documenting exceptions - the key to capturing exceptions is to focus on each process step from list of to-be process steps and ask the question "what could go wrong in this step?". If we apply this logic to the example from 3.5:
  - High level process step - Step #1
  - Event code - EXCEPTION\_101
  - Description - Unable to load configuration variables as network file share is down.
  - Actions:
    - Log the exception using the "03\_Log\_Activity" shared task bot.
    - Run the "03\_ExceptionMonitoring" task bot.
    - Exit the bot.