Contents

Introduction .................................................................................................................................................. 3
1. Name your Flow .................................................................................................................................. 3
2. Edit Flow Description ........................................................................................................................ 4
3. Consider Adding Co-Owners ............................................................................................................. 5
4. Create Power Automate Environments ............................................................................................... 5
5. Leverage Trigger Conditions ............................................................................................................. 6
6. Name Each Action .............................................................................................................................. 7
7. Add Comment to Actions .................................................................................................................. 7
8. Consider Retry Policy ........................................................................................................................ 8
9. Consider your Flow Connections ..................................................................................................... 9
10. Consider your Action Limits ........................................................................................................... 10
11. Consider your license Limits .......................................................................................................... 10
12. Use Variables to Refactor ............................................................................................................... 11
13. Flow Action Timeouts ..................................................................................................................... 11
14. Implement Error Handling ............................................................................................................. 12
Introduction

Power Platform enables Business users to transform themselves into Power users and thus feel empowered to quickly develop solutions for day to day needs. More and More business users are developing Power Automate flows, automating their day to day tasks and improving productivity.

Business need changes all the time. People Join and leave organizations. There is a very high chance that flow developed by you, will be supported or maintained by another co-worker in future and vice versa. Implementing Best practices would become an essential need to keep the business running as usual.

Purpose of this document is to articulate Best practices that can be adopted during Power Automate development.

1. Name your Flow

Whenever you create a new Flow, first thing you should do is to provide a Better name to it. Name should be simple and short to identify purpose of the flow. Few Examples are...

- Get SalesForce Data
- Sales Approval Process
- Request Approval Process
- Task Reminder Flow
- Budget Approval Process
- Transfer files to Shared Drive

In order to Change name of the flow, Click inside the default name once the flow is in edit mode and change it.
2. Edit Flow Description

Once you name the Flow, the Next step is to Edit Flow Description and provide details about the flow. This will help a new person to understand the purpose of the flow.

In order to edit Flow description, Click Edit in the details section of the Flow and edit the details.
3. Consider Adding Co-Owners

If the flow you are developing is for solving any business problem, it is good practice to add a co-owner to the flow. This can be either your colleague OR Your Boss OR A Service Account in case you use it. Goal here is to keep business process running by taking over, flows developed by an employee who left the organization.

To Add Co-Owner, click Owners Section within Flow details page and click Owners. Search for the owner name and add the user.

4. Create Power Automate Environments

You can create Flow Environments to control pipeline movements. It is best practice to always have at least 1 Non-Prod environment, where you can test your flow and once you find its working fine, deploy to Production Environment. Please refer following link to create Flow Environments...

https://docs.microsoft.com/en-us/power-automate/environments-overview-maker

In case you only have single environment within your company, it is recommended to create a flow with the name as QC\Dev first. Then test the flow and once you are satisfied, create a copy of the flow which you can use for Production Purpose.

To create a Copy of the flow, Click Save AS from Flow Details page and provide a new Name to the flow.
5. **Leverage Trigger Conditions**

For Certain Trigger Conditions, you have an option to configure Trigger Conditions. It is very handy and helpful option to only trigger your flow when condition is met. This will save your flow runs in case you have restriction based on your license.

Example: I have a Flow that triggers when a new Item is Saved in the List. But I want to run the flow only when Country is ‘United States’. So within trigger ‘When an item is created’, I can click Settings on Gear Icon of the Trigger and Enter following trigger condition.

![Settings for 'When an item is created'](image)

- **Split On**: Enable split-on to start an instance of the workflow per item in the selected array. Each instance can also have a distinct tracking id.
- **Array**: @triggerOutputs()[0]['body/value']
- **Custom Tracking Id**: Set the tracking id for the run. For split-on this tracking id is for the initiating request.
- **Secure Inputs (Preview)**: Secure inputs of the operation.
- **Secure Outputs (Preview)**: Secure outputs of the operation and references of output properties.
- **Retry Policy**: A retry policy applies to intermittent failures, characterized as HTTP status codes 408, 429, and 5xx, in addition to any connectivity exceptions. The default is an exponential interval policy set to retry 4 times.
- **Concurrency Control**: Limit number of concurrent runs of the flow, or leave it off to run as many as possible at the same time. Concurrency control changes the way new runs are queued. It cannot be undone once enabled.
- **Trigger Conditions**: Specify one or more expressions which must be true for the trigger to fire.
  - @equals(triggerBody)?['Country','United States']
6. Name Each Action

As soon as you add any action to the flow canvas, it is recommended to name the action to something meaningful. This way, another Flow Reader would know what this action does. At the same time, if you end up having many actions in the flow and would like to search a particular field in any action, searching and identifying in the field picker becomes an easier task. Best Practice is to always name every action.

To rename an action, Click on ... (ellipse) on the action Header and select rename. Following is an example of well-defined actions.

![Flow Diagram]

7. Add Comment to Actions

When you have a flow that has many different steps that do different things, it can sometimes be difficult to remember exactly what everything is for, and what you need to keep the flow working. Comments make it easy to annotate each individual action with notes to that you can easily remember what the flow needs:
8. Consider Retry Policy

A retry policy is another helpful setting that helps when there are intermittent failures to an Action. Most of the cases, default Type is already selected for Retry Policy. You can change it to specific interval based on your needs. When Specifying custom needs, you need to specify Intervals. Example as below.
9. Consider your Flow Connections

Many a times you end up having different or many connections within Flow. If you intend to use
Service accounts, make sure that you have correct connections being used within actions. Best way is to check Flow Summary page and Click connections. Then minimize the connections if there are duplicate connections. Also you can remove unused connections listed under Other connections. As we can see below, I have 1 used and 2 unused connections.

10. **Consider your Action Limits**
Another important practice to adapt is to know action limits so that you can design your flow using best possible approach. Example: Send an email Notification V2 can only send 100 emails in a day. In case you are not aware of this limit, your flow will fail to proceed further and action outcomes will be queued.

Best Practice is to go through Microsoft docs to understand the limits.

11. **Consider your license Limits**
Knowing your license is a key factor while designing flow. Refer below links to understand licensing FAQs.

https://docs.microsoft.com/en-us/power-platform/admin/powerapps-flow-licensing-faq

https://docs.microsoft.com/en-us/power-platform/admin/pricing-billing-skus

It is very important that you should know number of flow runs included in your license. This is a key indicator while designing your solution. Example – Office 365 V3 license includes Flow for Free which is a free license with limited number of flow runs. You must upgrade to Flow per user or per flow plan to get more Flow runs if solution demands.
12. **Use Variables to Refactor**
   You should consider refactoring where you find that common content is repeating across the flow. You can create a variable to hold common content.

   Example: I had to design a workflow which had 14 Approval actions. Almost 80% of the Email body of these actions were same. Refactoring this content into a variable and using it within the actions, is a best choice so that it is easy to change, in case needed.

13. **Flow Action Timeouts**
   There are timeouts for certain flow actions. Example: Approval Action has a default timeout of 30 days. It is always good to refer Flow Product Limits and configuration and design flow accordingly... [https://docs.microsoft.com/en-us/power-automate/limits-and-config](https://docs.microsoft.com/en-us/power-automate/limits-and-config)

   Use timeout settings to configure the timeouts... Example →

   ![Image of Flow Action Timeout Settings](image)
14. **Implement Error Handling**

Implementing Error Handling is also an important design considerations for certain requirements. If you expect to proceed further even after the earlier actions fail, you can capture the error and configure action settings accordingly. Example: As seen in the below flow, I wanted to proceed to get Budget List data OR Forecast List data if Actuals List Data fetch Fails. Dotted Line on the right shows a configuration which only runs if action fails. I am capturing Error and process proceeds further.

You need to configure flow Actions settings to run based on success\failures or Skip from the Flow Action ‘Configure to Run after’ page. In the below Image, you can see how I configured Get Budget List Data action based on earlier 2 actions...
Get Budget List data Action would execute if Applica to Each is successful OR Append Error to Actual List is successful. This way it can continue processing further.