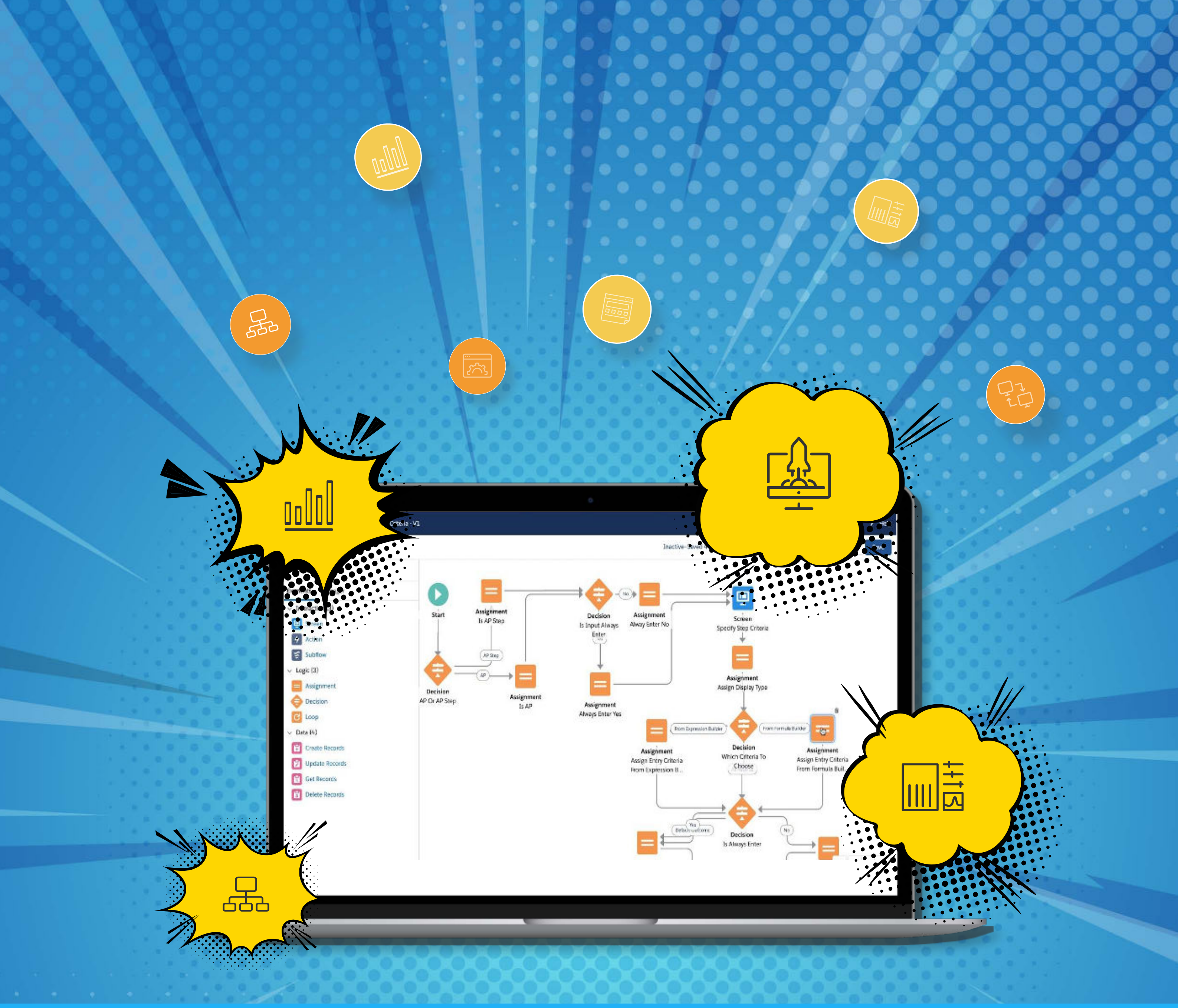


MIGRATING TO SALESFORCE FLOW

With Panaya

SFBEN | Panaya

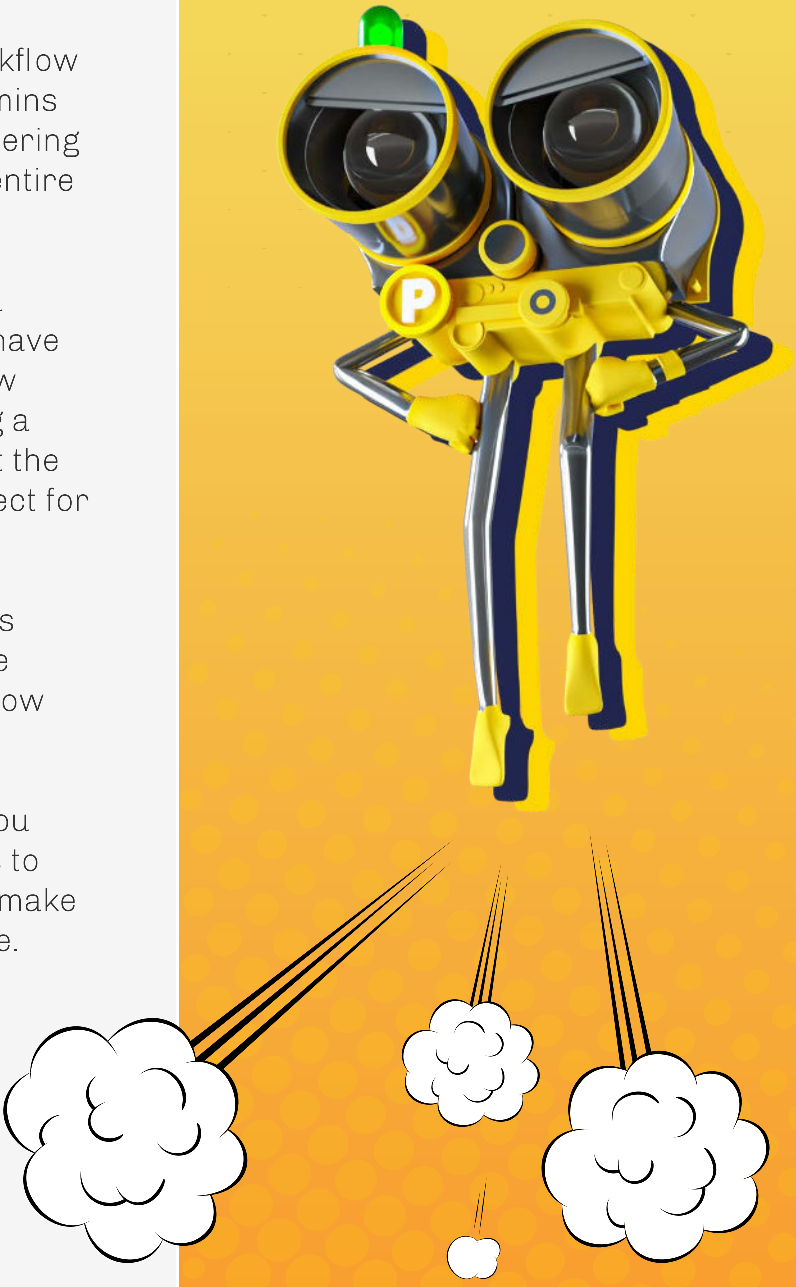
CONTENTS

GO WITH THE FLOW

With the imminent retirement of Workflow Rules and Process Builder, many admins may have spiraled into a panic, wondering how they are going to migrate their entire system to Salesforce Flow!

For many organizations, this will be a daunting task; for years, processes have been running on one system, and now everything has to be replicated using a different one. Flow is a great tool, but the migration is bound to be a huge project for Salesforce Admins.

We will cover the reasons why Flow is the declarative automation tool of the future, how to migrate to Flow, and how Panaya can help you achieve well-architected automations during the migration process. We'll also show you a 'day in the life' of someone who has to migrate, and provide top tips to help make your migration as smooth as possible.



LOW-CODE REVOLUTION

With Flow, anyone in an instance can implement custom logic to build a functional business system. It is a powerful tool that gives admins similar powers to developers. But why Flow over Process Builder and Workflow Rules?

The reasoning, at surface level, was to create a single tool that admins could use for any and all declarative automation requirements. Salesforce are moving their development away from coding, changing their approach to declarative automation – and Flow is the key component in this.

Plus, Flow includes a multitude of functionality that Workflow Rules and even Process Builder simply cannot do.

While your current Workflow Rules and Process Builders will continue to run, you will not be able to create new automation using these tools by the end of 2023, so migration is necessary and inevitable.

WHAT IS FLOW?

Flow Builder is the most powerful tool a Salesforce Admin has at their disposal. Performing mass updates across multiple unrelated records and putting complex logic into opportunity conversion are common examples of when you should use Flow.

There are three main 'building blocks' of any Flow:

1

Elements are the individual building blocks of the Flow. These perform logical actions such as assignments, decisions, or loops. There are also data elements that will query the database or commit record changes.

2

Connectors determine which element leads to which. Winter '21 enabled Auto-Layout, and connects the Elements together automatically.

3

Resources are the individual variables of data that are to be used in a Flow – these can be strings of text, numbers, records, formulae, or collections.

> *"The benefit of Salesforce Flow is that they are easy to maintain because anyone (assuming they know Flows) should be able to follow along with what you built."*
>
> Tim Combridge, Flow expert

Read more: The Complete Guide to Salesforce Flow

Why Do You Need It?

Flow is capable of so much more than either Workflow Rules or Process Builder, and the majority of parity gaps have been addressed.

In particular, Flow offers:

- Better overall **performance**.
- The **functionality** to improve high-volume automation such as Fast Field Updates (Before Save).
- Powerful **architecture** including error handling and debugging, and repeatable and reusable functionality such as the use of Sub-Flows.
- Additional features such as Screen Flows, and constant **development**.

Performance

Ultimately, Flow has faster performance and lower resource consumption than Workflow Rules and Process Builder – it's the better tool for the job.

Though you may be cursing this switch, in the long run you will be happy with the improved functionality, and so will your org! Before-Save Flows can improve performance by 10-20 times over Workflow Rules or Process Builder – if that's not a selling point then I don't know what is.

Functionality

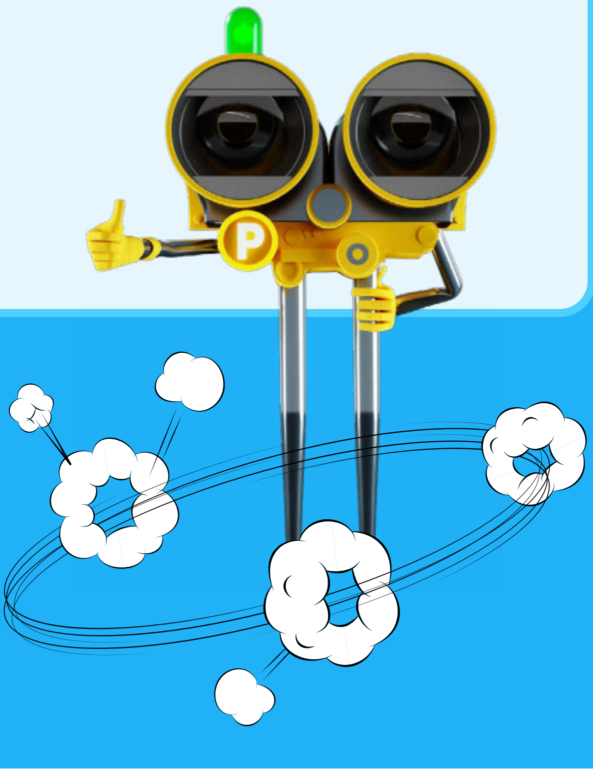
Flow includes a multitude of functionality that other legacy tools simply do not provide. It is the only declarative tool that offers the ability to handle multiple records at once in a collection, and it also supports looping (where a collection of variables or records can be iteratively processed one by one).

Architecture

Flow undeniably is far better architected to meet the increasing functionality and extensibility requirements of Salesforce users. It's a more fluid tool, meaning you can do a lot more with it than its predecessors.

It is not restricted to the "if this, then that" of Workflow Rule or "if this, then that, or that, or that..." of Process Builder. Those tools are limited by the need for existing relationships, whereas Flow Builder is not. With no such limitations, you can easily create a Flow to update many unrelated records to the one that has triggered the Flow to fire.

Flow Builder also gives you exceptional extensibility with Invocable Actions and Sub-Flows, for re-use around your org to empower more admins and standardize common interactions. The AppExchange has a dedicated catalog of Flow solutions and lightning components which can be added onto your flows to give a richer experience.



Future Development

Salesforce have made it clear that the future of no-code automation is Flow. At some point in the near future, you will no longer be able to create new Processes or Workflows. Presumably these platforms will be made redundant, so migrating to Flow will be inevitable.

With every release there are developments to Flow – Salesforce are constantly improving this tool, demonstrating their dedication and the trust they have in it.

Flow is crucial as the future of low-code technology, but we also realize organizations need well-architected implementation of automations. According to Salesforce, a well-architected org is organized to help you build solutions that are Trusted, Easy, and Adaptable:

- **Trusted** solutions protect stakeholders
- **Easy** solutions deliver value fast
- **Adaptable** solutions evolve with the business

When migrating your automations to Flow, make sure you follow these guidelines and ensure your new automations are aligned with Salesforce best practices. We recommend paying extra attention to the following suggestions:

- **Maintainability:** Your Flows should be simple to maintain and simple to investigate. It should be easy to detect the root cause of issues within the automation landscape.
- **Readability:** Ensure you can easily understand the structure of the Flow. Cohesive naming conventions, along with updated documentation, will help you achieve this goal.
- **Intentionality:** The Flows architecture goal is to provide business value. Keeping the structure aligned and focused on the business process, while removing technical debt, will help you deliver business value fast as you optimize your automations. It is also imperative to keep your Flow architecture ready for future enhancements.

The migration from Workflow Rules and Process Builder is an excellent opportunity to take a step toward a well architected and future-ready org.

HOW CAN PANAYA HELP?

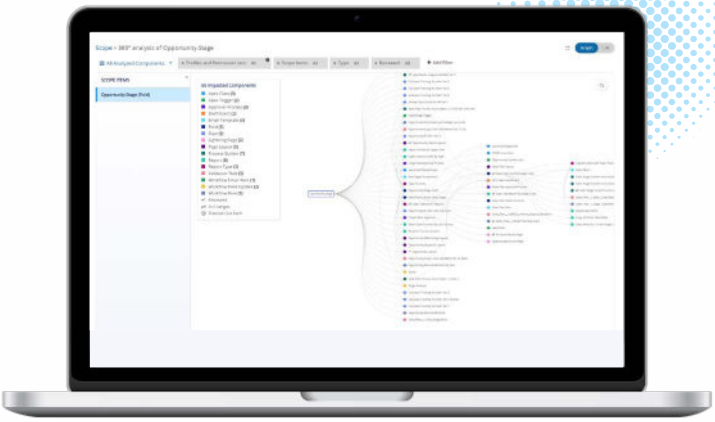
Panaya is a Change Intelligence platform that gives you real-time Impact Analysis with the click of a button – it has infrared vision inside your org. As you start your migration process, there are a few important stages where Panaya can help you...

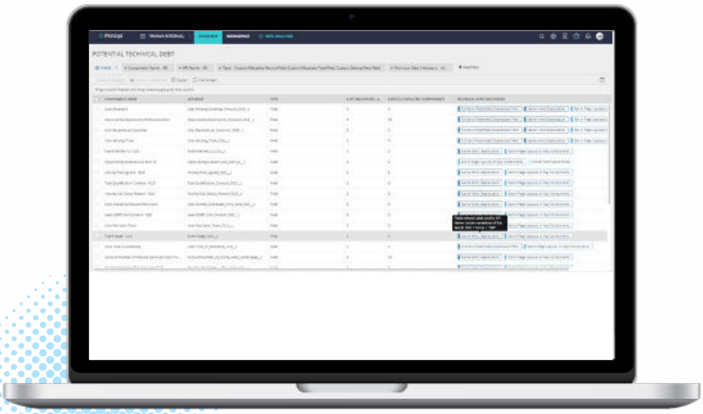
Analyze Your Automations

The journey towards a fruitful migration process starts with good planning. Make sure you understand your business goals, and that you have excellent knowledge of the current status of your automations.

Panaya gives you:

- Immediate visibility of your entire automations landscape.
- A clear understanding of the outcomes of each automation.
- An easy way to optimize your future architecture.
- Updated and accessible documentation of the different automations.



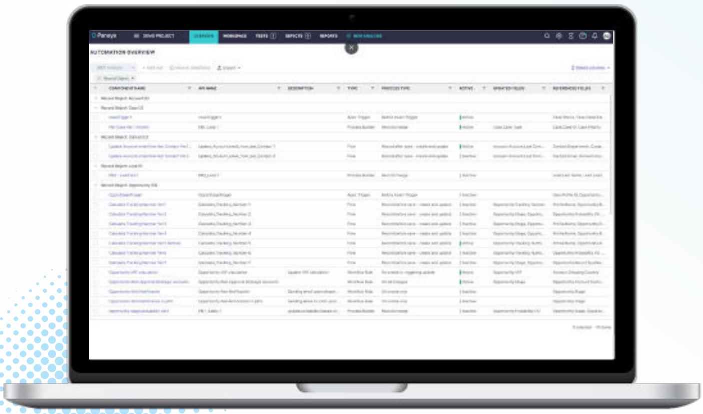


Identify and Remove Inactive Processes

As you start the migration process, take time to evaluate whether all existing processes are still needed.

Panaya can help you quickly identify inactive processes, as well as allow you to understand the impact of every automation before you delete or recreate it. This way you can ensure that you are not breaking any critical functionality and that you are cleaning technical debt safely – all without causing any disruption to the business.

Keep in mind that even after you finish the migration process, it is recommended to periodically clean inactive automations. With Panaya, you can export a report of inactive components, speeding up and reducing risk as you clean your org of technical debt.



Group All Automations by Objects

After cleaning inactive components, map all remaining automations and group them by object (standard or custom). By doing so, you will be able to see a clear view of all processes with the types and conditions, as well as any unique references to record type or profile names, and all the fields that are updated as a result of each process.

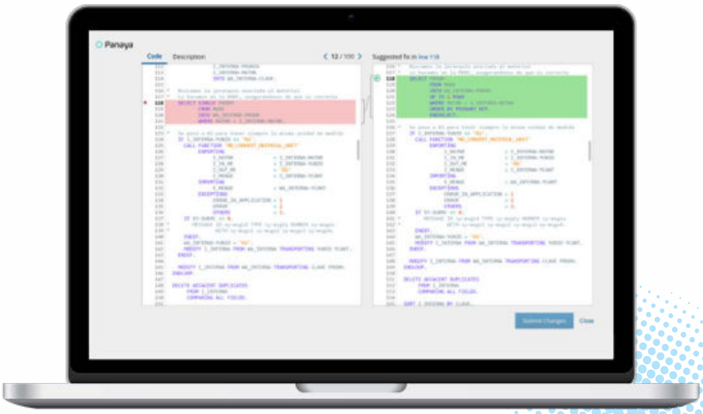
You can analyze the object automation's load and identify automations that can be removed or optimized with Flow. During this process, make sure you document all the information you need for creating your new Flows (object, criteria, updated fields). You will also have the option to export a list of critical automations filtered by object, type, or updated / referenced fields.

User Stories and Testing

After creating the new Flows, it is time to launch the business process execution and test its functionality.

While doing this, it is important to document all relevant steps. Proper documentation will not only help you and your team understand the reason behind every process, but will also help you in your audits, training, and onboarding of new team members. The evidence that is automatically attached to the user story or feature in Panaya's Test Management Module will allow you to accurately document all steps of the new business process.

You can launch "Test Recording" directly from each Flow entry to speed up your testing process and manage the UAT (User Acceptance Testing of your new Flows.



How is Panaya Different from the Salesforce Migration Tool?

Panaya provides a 360-degree view of the impact of change. This allows solution architects, admins, business analysts, and developers to gain visibility into any Salesforce change delivery before going into production to deliver customizations and change projects with confidence.

Panaya provides the full picture, giving you all of the information you require to implement Flows:

End-to-end process: Manage your migration in one place.

Zero-risk: Eliminate risk by applying change intelligence to predict impacts.

Optimized performance: Reduce technical debt.

Build for the future: Document your new automations.

> *With Panaya, organizations can streamline their migration to the new Salesforce technology. The goal is to take the complexity out of an intricate migration and map the entire org's automation, side effects, and dependencies. Then we can create an optimized version that is time-saving, clear, and built for the future of the business.*
>
> *Oz Lavee, CTO, Panaya, LoB*

A DAY IN THE LIFE...

OF ANDREW

Andrew has been tasked with migrating the entirety of an org's processes to Flow.
This is a huge task, but he has a plan to complete it smoothly and according to best practices.

He takes us through his thought process when migrating to Flow, and gives us his top tips for ensuring a successful project.

Make Time

First thing's first, you need to set out some time specifically for this project. I tend to work on an object at a time, and plan my time accordingly. For example, one day I would work on automations for the Account object, another day on the Contact object.

Something I had to overcome with this project is distractions – if you aren't totally focused it can quickly go wrong. I utilize the "Focus" feature on my Apple devices, putting them on "Do Not Disturb" for the time I need to dedicate to the task at hand. Having no email or Slack messages to distract me means I can engross myself in what needs to be done. Also, blocking out my calendar for designated Flow builder time ensures no one tries to schedule any meetings.

Know Your Org

You need to be an expert in your org to take on a project like this. It's not enough to just know which objects you use, you need to know the relationships, all processes for everything you use (sales/service/marketing), the other tools being used, the integrations you have, and the data structure. Without knowing your org inside out, you may encounter issues with processes you weren't aware of.

Document Your Automations

I start by mapping out the automations for a particular object, for example, all of the Workflow Rules and Process Builders for the Account object. I then take into consideration any automations that are on another object but may have an impact on the Account object, like a Process Builder for the Opportunity object that would update related account records.

Map Out Where Automations Can Be Merged

Flow Builder is so much more powerful than Workflow Rules and Process Builder. Because of this, you will discover that certain individual automations you built in these tools can be combined in Flow Builder.

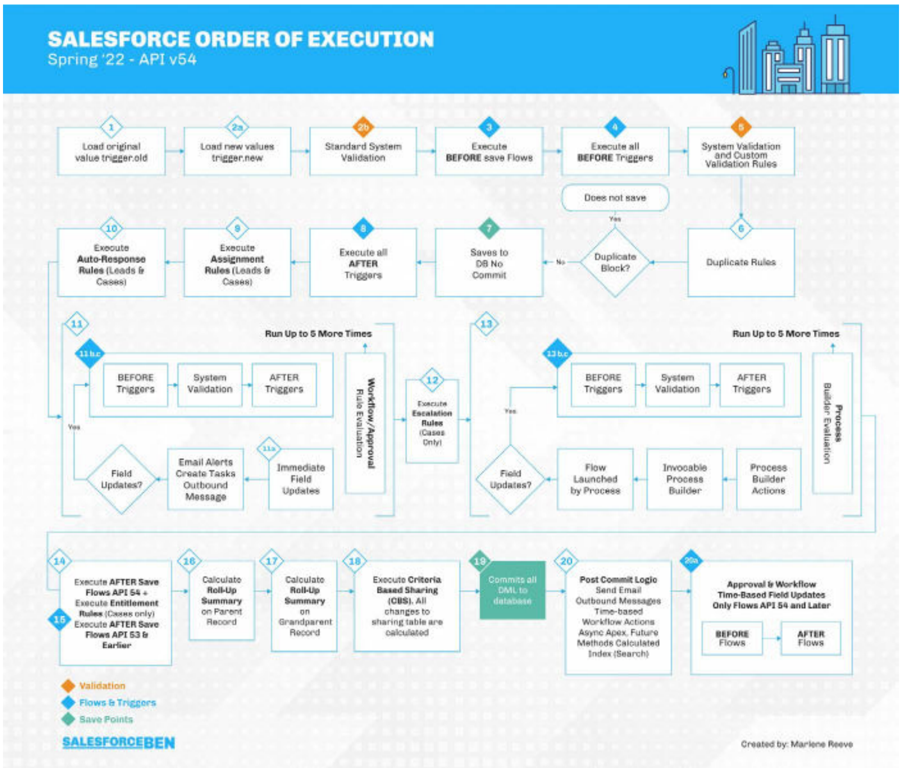
Here's a good example of this; my org had a number of individual Workflow Rules to assign accounts to users based on their territory and the revenue of the account. These were easily combined into one single Flow, and doing this greatly improved the performance of the overall sales system.

The Order of Execution

This is a lot more important than many people realize. I reference the diagram below when refactoring my automations – it gives me a greater understanding and appreciation of the platform.

When documenting my automations, I put a number next to each one in reference to where it comes in the order of execution. In the example above, the account assignment Workflow Rules would be 13 in the order of execution.

I then do the same thing with the new refactored automation when mapping out where automations could be merged. This allows me to note that with the new account assignment Flow, what was 13 in the order of execution would now be 3 because of the Before-Save Flow being used. You can then paint a bigger picture in terms of your overall process, as you can identify what could potentially cause issues.


SALESFORCE ORDER OF EXECUTION
Spring '22 - API v54
1
Load original value trigger.old
2a
Load new values trigger.new
2b
Standard System Validation
3
Execute BEFORE save Flows
4
Execute all BEFORE Triggers
5
System Validation and Custom Validation Rules
6
Duplicate Rules
Duplicate Block?
Yes
Does not save
No
7
Saves to DB No Commit
8
Execute all AFTER Triggers
9
Execute Assignment Rules (Leads & Cases)
10
Execute Auto-Response Rules (Leads & Cases)
11
Run Up to 5 More Times
11 b.c
BEFORE Triggers
System Validation
AFTER Triggers
Workflow/Approval Rule Evaluation
11a
Immediate Field Updates
Email Alerts Create Tasks Outbound Message
Field Updates?
Yes
12
Execute Escalation Rules (Cases Only)
13
Run Up to 5 More Times
13 b.c
BEFORE Triggers
System Validation
AFTER Triggers
Process Builder Evaluation
Process Builder Actions
Invocable Process Builder
Flow Launched by Process
Field Updates?
Yes
14
Execute AFTER Save Flows API 54 + Execute Entitlement Rules (Cases only)
15
Execute AFTER Save Flows API 53 & Earlier
16
Calculate Roll-Up Summary on Parent Record
17
Calculate Roll-Up Summary on Grandparent Record
18
Execute Criteria Based Sharing (CBS). All changes to sharing table are calculated
19
Commits all DML to database
20
Post Commit Logic Send Email Outbound Messages Time-based Workflow Actions Async Apex, Future Methods Calculated Index (Search)
20a
Approval & Workflow Time-Based Field Updates Only Flows API 54 and Later
BEFORE Flows
AFTER Flows
Validation
Flows & Triggers
Save Points
SALESFORCEBEN
Created by: Marlene Reeve

Test Your Flows

I knew that by combining a lot of individual automations into a single Flow, I would need to thoroughly test what I was building. Not only do I need to make sure it works as expected, I also need to make sure everything else in the overall process is still working. Cue, testing in Flow. The testing part can be divided into two parts.

First: Testing the Flow as a unit. Here, I can make use of the new built-in testing and debug tool to test my Flows. It allows me to create tests in a similar way to creating test classes with Apex. I set the test details, what would trigger the test, the path, the initial triggering record field values, and the test assertions. These are all saved within the Flow – when moving the Flow between sandbox environments, I am able to use the same tests I created.

Second: End-to-end testing and the business users' validation. Our goal here is to make sure that the business process continues to work as expected and that the business users' activities will not be affected. For this part you can use Panaya. It helps to manage the testing efforts end-to-end and guarantees a seamless handshake with the business users. Panaya will automatically document the business users' feedback and create evidence of the process for audits, as well as future rollout activities.

TIPS FOR SUCCESS

1. Standardize your naming conventions for Flows, Elements, Variables, etc. This makes them easier to work with, especially if you have to rework them or if someone else joins the company.

2. Ensure there are no pink elements (get, create, update, delete) in Loops.

3. Ensure thorough documentation to understand why you built the Flow and what exactly you built.

4. Split your Flows into the smallest possible chunks.

5. Make your flows reusable and scalable.

6. Be ready for errors.

7. **Test, test, test!**

FINAL THOUGHTS

With Flow, anyone in an instance can implement custom logic to build a functional business system. It is a powerful tool which gives admins similar powers to developers. Though migrating to Flow will inevitably be a huge task, it is both necessary and unavoidable as Salesforce moves towards declarative automation.

Using the four key steps (**map, clean, implement**, and **test**) your migration to Flow should be more straightforward, **but Panaya can make this move even easier.**

Don't forget to book a product demo to learn more about how Panaya can help you migrate to flow.

BOOK A DEMO