Reuse your Flows with Subflow

Salesforce Women in Tech Group - Columbia, MD, USA
https://katiekodes.com/subflow-witmd23/

What is Flow Builder?

Flow Builder is a point-and-click tool for building flowchart-based code-like logic (“Flows”) that do useful things in Salesforce.

There are 3 main types:

1. Screen
2. Triggered
3. Autolaunched

#1: Screen Flows

- Run interactively.
- Run in response to human clicks & typing.

DEMO: Screen Flow

#2: Triggered Flows

- Run in the background.
- Background. Via schedule, data edits, “platform events.”Interactive.
- Run in response to:
  - Schedules, or
  - Data edits, or
  - “Platform events”

DEMO: (Record)-Triggered Flow

#3: Autolaunched Flows

- Run in the background.
- You don’t think about WHAT they’ll run in response to when you build them. (Most commonly, though, they get run from…)
  - other Flows (then it’s a “**subflow**!” That’s all “subflow” means.)
  - Apex code

Why Subflow / Autolaunched Flow?

To design with “modularity” *(small, specialized logic clumps)*:

> “All cooking knives should live in a knife block, not mixed with the spoons.”

1. Tidiness / Readability
2. Maintenance / Reusability
    - Edit once, update everywhere

Extreme Subflowing: “Build **every** Flow first as autolaunched”

- Counterpoint: “D.R.Y. premature optimization”
- Counterpoint: (Fast Field Update record-triggered flows get tricky)

More “modularity” / reusability besides Subflow

- Fetching data (custom metadata, custom settings, or Salesforce object data)
- Packaging
  - (If you’ve ever installed something from the Unofficial Salesforce blog, that’s what they do)

DEMO: Subflow

Redesign 2 Flows into 3 Flows
(2 parent Flows both calling 1 shared child Subflow)

Recognizing Subflow (“modularity”) opportunities

1. ***Copying & pasting*** inside/between Flows
2. ***Save As*** from another Flow
3. ***If-Then-Else branching*** with nearly-identical branches

I ❤️ paper & 25 colors of highlighter play.

- 👍 “DRY” Flows: “Don’t Repeat Yourself”
- 👎 “WET” Flows: “Write Everything Twice” 😂

Avoiding overthinking about “DRY:”

- 1 redundancy = life happens
- 2+ redundancies = time to Subflow

Advanced Subflow

More “Refactoring,” even better “Decoupling”

Advanced subflow: “services”

Not all Autolaunched Flows / Subflows are “services.”

(Not all tasty cakes are strawberry!)

Only the ones that ***make no assumptions*** about ***how*** they’ll be “called” are.

1. They ***never edit*** data.
2. They avoid “record”-typed ***input*** variables (stick to text / number / etc.)
    - (Except some Custom Metadata & other very-unique “config” record types can be OK)
3. They always have >= 1 ***output*** variable(s).

“Service” subflow benefits

1. Reuse an Autolaunched Flow in more kinds of Subflow + outside of Flow.
    - (A street-address helper should be reusable for Lead, Contact, Account, or no record at all.)
        - from a Screen Flow
        - from a Record-Triggered Flow
        - from an API-driven “find your sales rep” public website that doesn’t edit any data at all
2. Tidier testing. Separate out thinking about:
    - “Does it suggest a value correctly?” from
    - “Does it post the value to Chatter?” / “Is the value displayed beautifully in Screen Flow?”

Thanks!
Questions?

https://katiekodes.com/subflow-witmd23/