

✓ DO

✓ **Entry Conditions immer setzen**

Ohne Conditions läuft der Flow bei jeder Änderung am Record – auch bei irrelevanten Feld-Updates. Definieren Sie exakt, wann der Flow feuern soll.

✓ **Fault Paths einrichten**

Jeder DML- und Callout-Knoten braucht einen Fault Path. Senden Sie Fehler per E-Mail an den Admin – lautlose Fehler sind die schlimmsten.

✓ **Subflows für wiederverwendbare Logik**

DRY-Prinzip: Gleiche Logik gehört in einen Subflow, nicht in fünf kopierte Versionen. Änderung an einer Stelle, Wirkung überall.

✓ **Naming Convention durchhalten**

[Typ] [Objekt]: [Zweck]. Beispiel: „RTF Account: Set Region on Create“. Gilt auch für Variablen und Elemente im Flow.

✓ **Flow-Beschreibung ausfüllen**

Was tut der Flow, wann läuft er, warum existiert er? Das Description-Feld ist kein Nice-to-have – es ist Dokumentation für Ihr zukünftiges Ich.

✓ **Bulk-fähig bauen**

Loops + Collections statt Einzelaktionen. Testen Sie mit 200 Records – sonst brechen Governor Limits beim nächsten Data Load.

✓ **Before Save für Feld-Updates nutzen**

Felder auf dem gleichen Record setzen? Before Save spart einen DML-Call und ist deutlich performanter als After Save + Update.

✗ DON'T

✗ **Kein DML in Loops**

SOQL/DML-Limits brechen garantiert. Daten in einer Collection sammeln, dann ein einziger DML-Call außerhalb des Loops.

✗ **Keine hartkodierten Record IDs**

Bricht bei jeder Sandbox-Refresh und Migration. Custom Metadata Types oder Custom Settings sind die richtige Alternative.

✗ **Nicht „Created or Updated“ ohne Condition**

Läuft bei jeder Änderung, auch irrelevanten. Performance-Killer und Fehlerquelle Nr. 1 in jeder Org.

✗ **Keinen Process Builder neu anlegen**

Flows sind der Standard seit Spring '22. Bestehende Process Builder migrieren, keine neuen erstellen. Salesforce stellt den Support ein.

✗ **Keine toten Flows liegen lassen**

Inaktive Flows und alte Versionen aufräumen. Jede Flow-Leiche kostet Übersicht und erhöht die Deployment-Komplexität.

✗ **Keine Flows ohne Test in Production**

Immer in der Sandbox testen – mit realistischen Daten und Edge Cases. Flow-Fehler in Prod treffen sofort alle User.

✗ **Mehrere RTFs pro Objekt vermeiden**

Die Ausführungsreihenfolge ist nicht garantiert. Ein RTF pro Objekt und Trigger-Zeitpunkt, Subflows für Verzweigungen.

⚠ **Flow schlägt fehl? Schnell-Checkliste**

- 1 Debug Log prüfen →
- 2 Fault Path vorhanden? →
- 3 Entry Condition korrekt? →
- 4 Bulk-Test mit 200 Records →
- 5 Rekursion ausschließen

★ **Profi-Tipps aus der Praxis**

- › Flow Trigger Explorer zeigt alle Automations pro Objekt auf einen Blick
- › Custom Labels für Texte in Screen Flows – änderbar ohne Deployment
- › \$Record.LastModifiedById prüfen, um Flow-on-Flow-Loops zu verhindern
- › Debug-Flow-Interviews in Setup nutzen statt Print-Debugging mit Assignments