📄 Phase – 2: Backend Development & Configurations

Automation Logic

Description

The Automation Logic phase focuses on automating the end-to-end lifecycle of the **Software Installation Request** using ServiceNow's workflow capabilities. The automation ensures that once a user submits a software request through the Service Catalog, the request is routed for approval, fulfillment tasks are created for the IT team upon approval, and appropriate request updates are handled in case of rejection.

Although the SkillWallet instructions reference the legacy **Workflow Editor**, the automation logic for this project has been implemented using **Flow Designer**, which is the modern and recommended ServiceNow approach for workflow automation. The same business logic and outcomes are achieved in a more scalable and maintainable manner.

Activity 1: Creation of Workflow

Objective

To create a workflow that automates the processing of software installation requests submitted via the Service Catalog.

Implementation Approach

The workflow was created using **Flow Designer**, triggered by Service Catalog submissions. The flow is configured to operate on the **Requested Item (RITM)** record, which serves as the central record for approvals and fulfillment activities.

Steps Performed

1. Logged into the ServiceNow instance.

2. Clicked on **All** and searched for **Workflow Studio**.

3. Opened **Workflow Studio**.

4. Clicked on **New** → **Flow**.

5. Entered the following details:

    - **Flow Name**: Software Request Workflow

    - **Application**: Global

6. Clicked **Create / Submit** to initialize the flow.

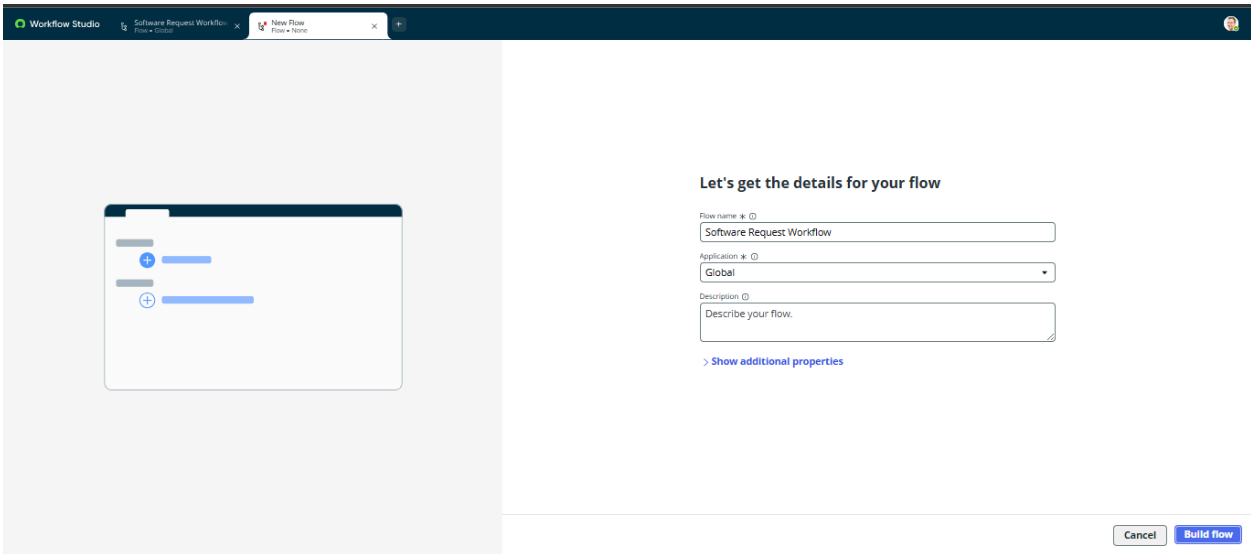
Trigger Configuration

7. Added a trigger of type **Service Catalog**.

8. Configured the trigger to run when a **Service Catalog item is requested**, creating a Requested Item (RITM).

9. Used the **Requested Item record** from the trigger as the primary data source for subsequent actions.

📸 **Screenshot Evidence – Activity 1**
Insert screenshot showing the creation of the flow and trigger configuration (Workflow Studio / Flow Designer setup screen).

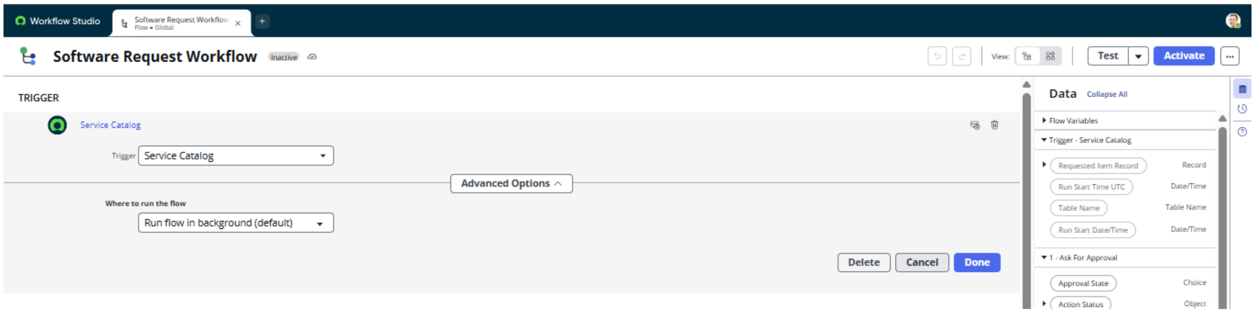


Let's get the details for your flow

Flow name *: Software Request Workflow

Application *: Global

Description: Describe your flow.



Trigger: Service Catalog

Where to run the flow: Run flow in background (default)

Trigger - Service Catalog

Requested Item Record	Record
Run Start Time UTC	Date/Time
Table Name	Table Name
Run Start Date/Time	Date/Time

1 - Ask For Approval

Approval State	Choice
Action Status	Object

Activity 2: Workflow Design and Automation Logic

Objective

To design the automation logic that handles approvals, task creation, and request updates based on approval outcomes.

Workflow Design Overview

The workflow follows a structured sequence:

- Triggered on Service Catalog submission
- Routes the request for approval

- Creates a fulfillment task upon approval
- Updates the Requested Item appropriately if rejected
- Ends the flow cleanly in all scenarios

Step 1: Approval Configuration

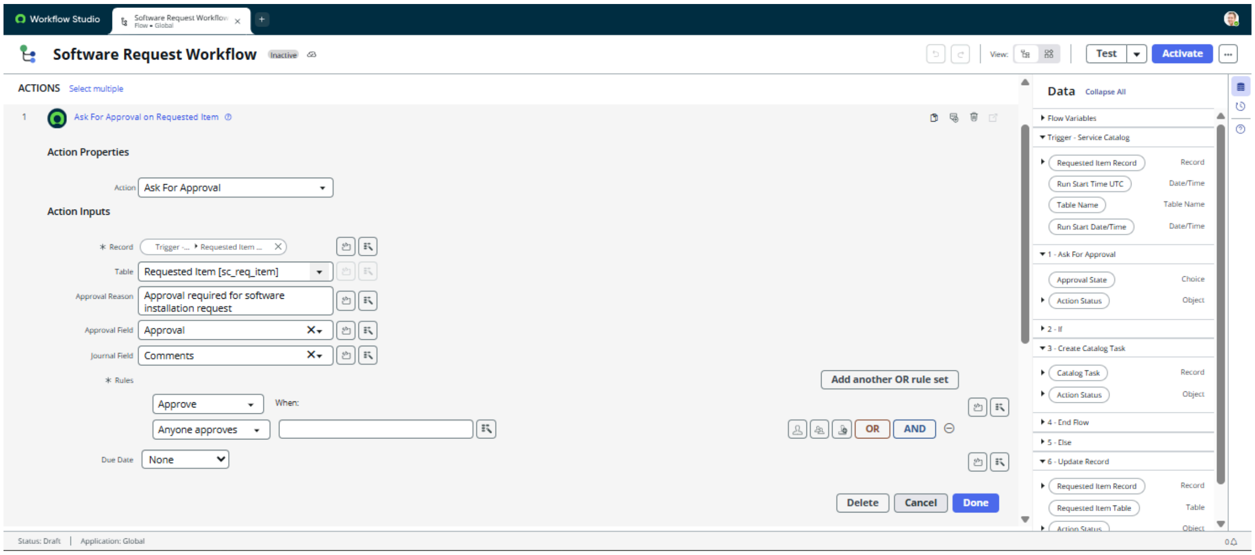
An approval action was added to the flow to request authorization for the software installation.

- **Action Used**: Ask For Approval on Requested Item
- **Record**: Requested Item (from trigger data)
- **Approval Reason**: Software installation approval
- **Approval Field**: Approval

This step ensures managerial or designated approval before proceeding with fulfillment.

📸 Screenshot Evidence

Insert screenshot showing the "Ask For Approval on Requested Item" action configuration.



Software Request Workflow Inactive

ACTIONS

1 Ask For Approval on Requested Item

Action Properties

Action: Ask For Approval

Action Inputs

* Record: Trigger -... ▸ Requested Item ...
Table: Requested Item [sc_req_item]
Approval Reason: Approval required for software installation request
Approval Field: Approval
Journal Field: Comments
* Rules: Approve When: Anyone approves
Due Date: None

Data

- ▸ Flow Variables
- ▾ Trigger - Service Catalog
  - Requested Item Record — Record
  - Run Start Time UTC — Date/Time
  - Table Name — Table Name
  - Run Start Date/Time — Date/Time
- ▾ 1 - Ask For Approval
  - Approval State — Choice
  - Action Status — Object
- ▸ 2 - If
- ▾ 3 - Create Catalog Task
  - Catalog Task — Record
  - Action Status — Object
- ▸ 4 - End Flow
- ▸ 5 - Else
- ▾ 6 - Update Record
  - Requested Item Record — Record
  - Requested Item Table — Table
  - Action Status — Object

Status: Draft | Application: Global

Step 2: Conditional Logic (Approval Outcome)

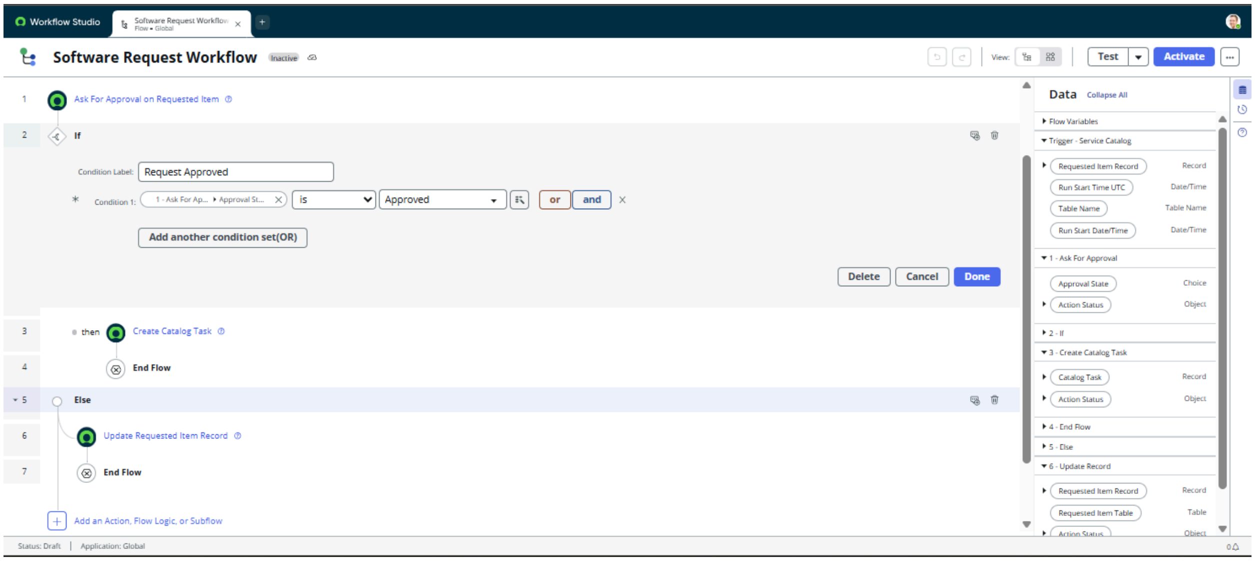
A conditional logic block was added to evaluate the approval result.

- **Condition**: Approval State is Approved

This IF condition determines whether the workflow should proceed with fulfillment or handle rejection.

📸 Screenshot Evidence

Insert screenshot showing the IF Request Approved


Workflow Studio
Software Request Workflow
Flow • Global
+
Inactive
View:
Test
Activate
Add another condition set(OR)
or
and
Delete
Cancel
Done
Add an Action, Flow Logic, or Subflow
Data
Collapse All

Software Request Workflow

1 Ask For Approval on Requested Item

2 If

Condition Label: Request Approved

* Condition 1: 1 - Ask For Ap... ▸ Approval St... | is | Approved

3 then Create Catalog Task

4 End Flow

5 Else

6 Update Requested Item Record

7 End Flow

Status: Draft | Application: Global

Data

▸ Flow Variables

▾ Trigger - Service Catalog

Requested Item Record	Record
Run Start Time UTC	Date/Time
Table Name	Table Name
Run Start Date/Time	Date/Time

▾ 1 - Ask For Approval

Approval State	Choice
Action Status	Object

▸ 2 - If

▾ 3 - Create Catalog Task

Catalog Task	Record
Action Status	Object

▸ 4 - End Flow

▸ 5 - Else

▾ 6 - Update Record

Requested Item Record	Record
Requested Item Table	Table
Action Status	Object

Step 3: Fulfillment Task Creation (Approved Path)

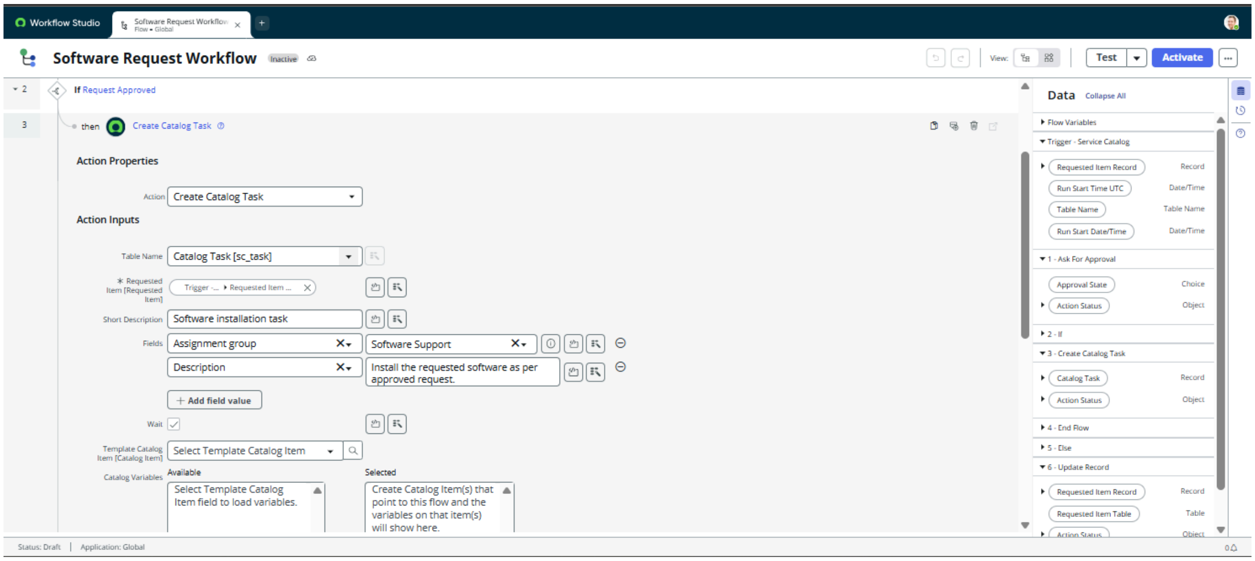
If the request is approved, the workflow automatically creates a Catalog Task for the fulfillment team.

- **Action Used**: Create Catalog Task
- **Requested Item**: Passed from trigger
- **Short Description**: Software installation task
- **Description**: Install the requested software as per approved request
- **Wait for Completion**: Enabled

This ensures that the workflow pauses until the task is completed, maintaining process integrity.

📸 Screenshot Evidence

Insert screenshot showing the Create Catalog Task action configuration.



Software Request Workflow Inactive

2 If Request Approved

3 then Create Catalog Task

Action Properties

Action: Create Catalog Task

Action Inputs

Table Name: Catalog Task [sc_task]

* Requested Item [Requested Item]: Trigger -... ▸ Requested Item ...

Short Description: Software installation task

Fields: Assignment group — Software Support

Description — Install the requested software as per approved request.

Wait: ☑

Template Catalog Item [Catalog Item]: Select Template Catalog Item

Catalog Variables: Available — Select Template Catalog Item field to load variables. Selected — Create Catalog Item(s) that point to this flow and the variables on that item(s) will show here.

Status: Draft | Application: Global

Step 4: Rejection Handling (Else Path)

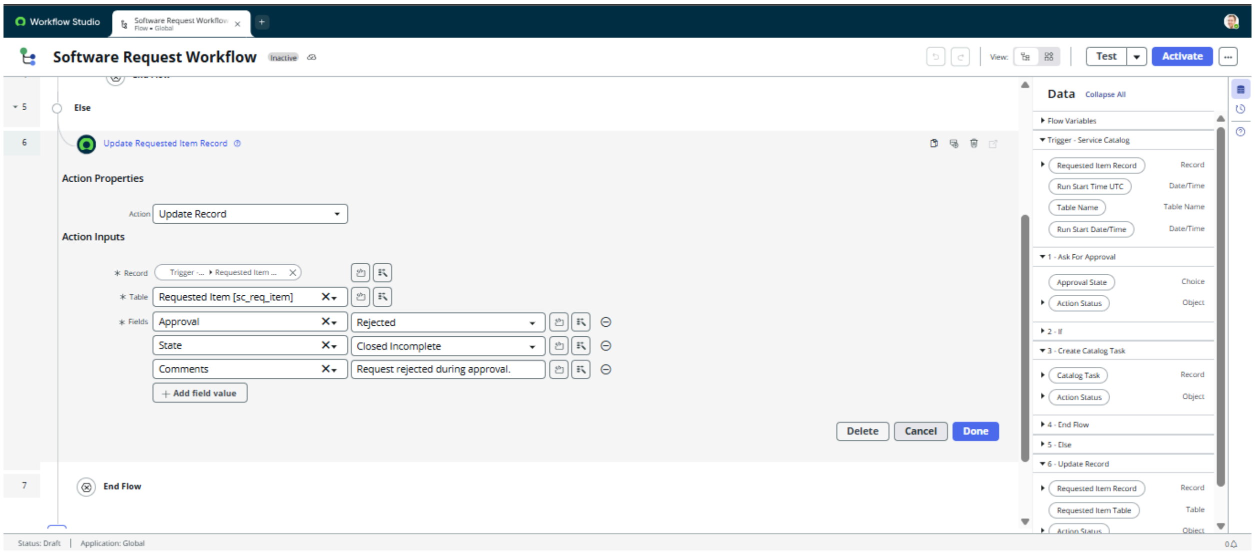
If the request is not approved, the workflow updates the Requested Item record.

- **Action Used**: Update Record
- **Table**: Requested Item [sc_req_item]
- **Update**: Request state or approval status updated to reflect rejection

This ensures that rejected requests are properly tracked and closed without creating fulfillment tasks.

📸 Screenshot Evidence

Insert screenshot showing the Else If Request Rejected



5 Else

6 Update Requested Item Record

Action Properties

Action: Update Record

Action Inputs

* Record: Trigger - ... ▸ Requested Item ...
* Table: Requested Item [sc_req_item]
* Fields: Approval — Rejected
State — Closed Incomplete
Comments — Request rejected during approval.

+ Add field value

7 End Flow

Flow Variables
Trigger - Service Catalog

Requested Item Record	Record
Run Start Time UTC	Date/Time
Table Name	Table Name
Run Start Date/Time	Date/Time

1 - Ask For Approval

Approval State	Choice
Action Status	Object

2 - If

3 - Create Catalog Task

Catalog Task	Record
Action Status	Object

4 - End Flow

5 - Else

6 - Update Record

Requested Item Record	Record
Requested Item Table	Table
Action Status	Object

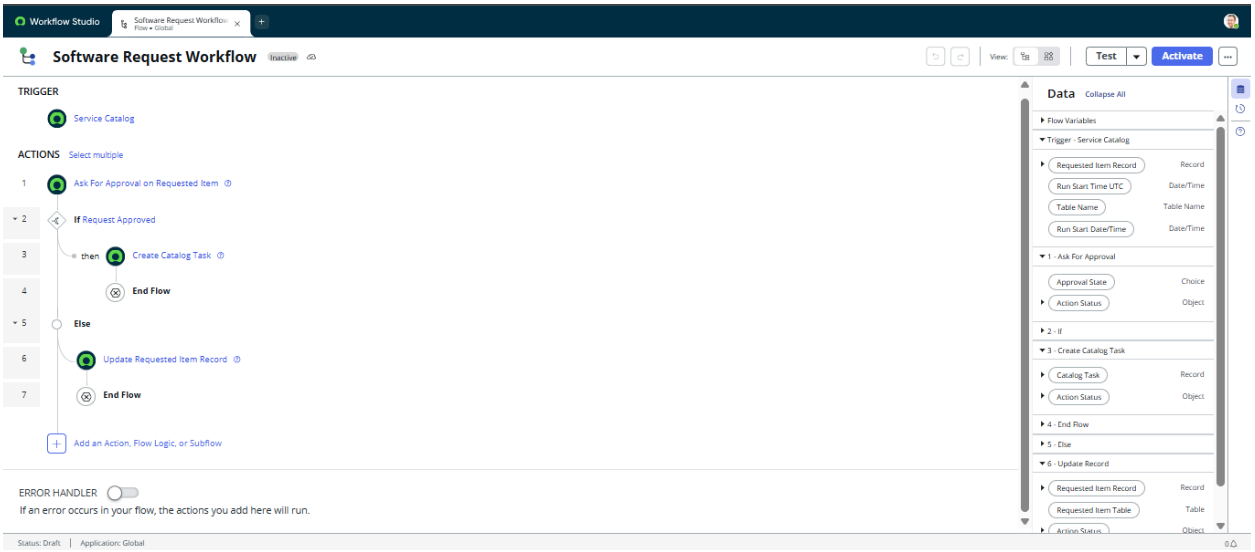
Status: Draft | Application: Global

Step 5: Workflow Completion

End Flow actions were added to both approval and rejection paths to clearly mark workflow completion and avoid unintended execution beyond defined logic.

📸 Screenshot Evidence – Final Workflow

Insert screenshot of the complete Flow Designer canvas showing approval, condition, task creation, update record, and end flow (your final flow – the 7th screenshot you mentioned).



TRIGGER

Service Catalog

ACTIONS Select multiple

1 Ask For Approval on Requested Item

2 If Request Approved

3 then Create Catalog Task

4 End Flow

5 Else

6 Update Requested Item Record

7 End Flow

Add an Action, Flow Logic, or Subflow

ERROR HANDLER

If an error occurs in your flow, the actions you add here will run.

Flow Variables
Trigger - Service Catalog

Requested Item Record	Record
Run Start Time UTC	Date/Time
Table Name	Table Name
Run Start Date/Time	Date/Time

1 - Ask For Approval

Approval State	Choice
Action Status	Object

2 - If

3 - Create Catalog Task

Catalog Task	Record
Action Status	Object

4 - End Flow

5 - Else

6 - Update Record

Requested Item Record	Record
Requested Item Table	Table
Action Status	Object

Status: Draft | Application: Global

Validation and Observations

The workflow was tested by submitting a Software Installation Request through the Service Catalog. The following behaviors were validated:

- Approval request generated automatically upon submission
- Catalog Task created only after approval
- Requested Item updated correctly on rejection
- Workflow executed end-to-end without errors

Outcome

The Automation Logic successfully implements a robust and scalable workflow for handling software installation requests. By leveraging **Flow Designer**, the solution ensures modern automation practices, improved maintainability, and alignment with ServiceNow best practices. This automation significantly reduces manual intervention, improves turnaround time, and provides a consistent experience for requesters, approvers, and fulfillment teams.