



Phase 4: Data Migration, Testing & Security

Data Integrity

Description

The Data Integrity phase focuses on ensuring that all information related to the Software Installation Request is accurate, complete, secure, and consistently maintained throughout the request lifecycle. This phase validates mandatory field enforcement, automated data population, approval controls, and end-to-end workflow behavior to prevent incomplete or unauthorized submissions.

Activity 1: Mandatory Field Enforcement

Objective

To ensure that critical software request details are mandatory and cannot be bypassed during submission.

Actions Taken

- Key fields such as:
 - **Software Name**
 - **Version (if required)**
 - **Business Justification**
 - **Urgency Level**
- were configured as **mandatory**.
- Client-side validation was implemented using **Catalog Variable settings and UI behavior**.

- Server-side enforcement was ensured through **workflow-driven approval checks**, preventing incomplete records from progressing.

Outcome

- Users were unable to submit requests with missing mandatory information.
- Incomplete or invalid requests were blocked at the submission stage, ensuring clean data entry.

Screenshot Evidence (Optional)

Insert screenshot showing mandatory fields marked with red asterisks () on the Service Catalog form.*

The screenshot shows the 'Software Installation Request' form in the Service Catalog. At the top, there's a breadcrumb navigation: Home > Service Catalog > Software > Software Installation Request. A search bar labeled 'Search Catalog' is on the right. The main form has several input fields:

- A note: 'Request installation of licensed software'.
- A required field: 'What software do you need?' with the value 'MS Word, IntelliJ IDEA'.
- A note: 'Specify version (if required)' with the value '2021, 17.0.8'.
- A note: 'Why do you need this software?' with the text 'This software is required to perform my daily project tasks efficiently. It is needed to meet current project requirements and improve productivity.'
- A note: 'Select urgency level.' with options: Normal (selected), High, and Critical.
- An optional field: 'Add attachments'.
- On the right side, there are buttons for 'Quantity' (set to 1), 'Delivery Time' (0 Days), 'Add to Cart', 'Save as Draft', and a large blue 'Order Now' button.
- A sidebar on the right lists 'Required information' with two red asterisked fields: 'What software do you need?' and 'Why do you need this software?'.

Activity 2: Automatic Population of Values

Objective

To reduce manual input errors by automatically populating user-related data.

Actions Taken

- The **Requested For** field was auto-populated with the logged-in user.

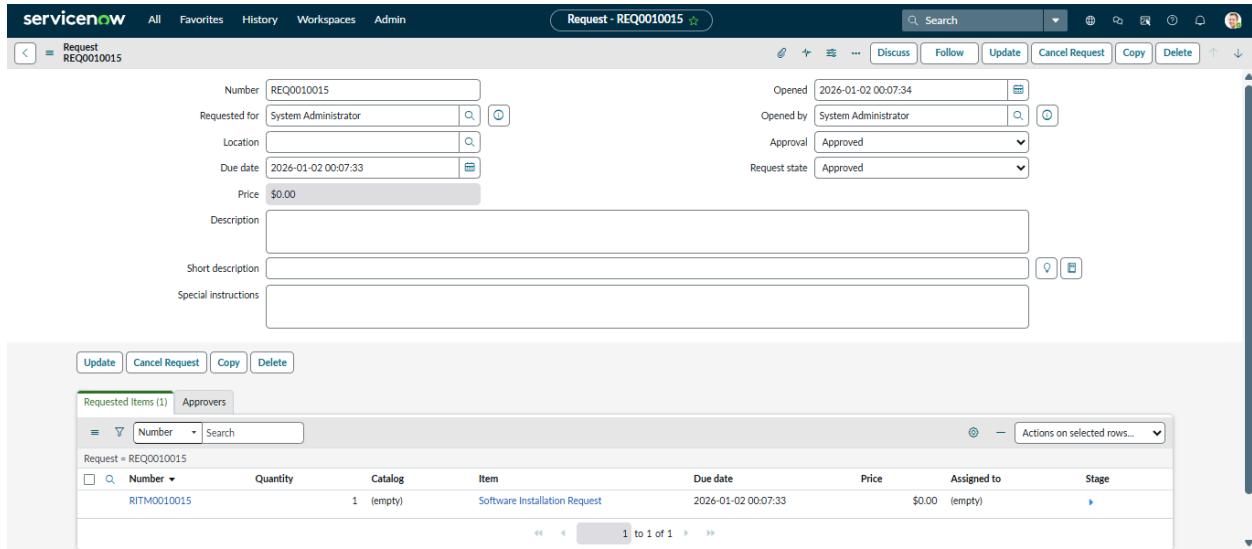
- User-related attributes such as **Department** and **Location** were dynamically pulled from the user profile.
- Default values were applied where applicable to maintain consistency.

Outcome

- Reduced manual effort for users.
- Ensured consistency and accuracy of user-related data across records.

Screenshot Evidence (Optional)

Insert screenshot showing auto-filled Requested For and related user fields in the Request or RITM record.



The screenshot shows the ServiceNow interface for a Request record (REQ0010015) and its associated RITM record (RITM0010015).

Request Record (REQ0010015):

Number	REQ0010015	Opened	2026-01-02 00:07:34
Requested for	System Administrator	Opened by	System Administrator
Location		Approval	Approved
Due date	2026-01-02 00:07:33	Request state	Approved
Price	\$0.00		
Description			
Short description			
Special instructions			

RITM Record (RITM0010015):

Number	Quantity	Catalog	Item	Due date	Price	Assigned to	Stage
RITM0010015	1 (empty)	Software Installation Request		2026-01-02 00:07:33	\$0.00 (empty)		

Activity 3: Testing & Validation

Objective

To validate the complete lifecycle of a Software Installation Request.

Steps Performed

- Submitted a Software Installation Request from the Service Catalog.
- Verified approval flow execution.
- Confirmed catalog task creation upon approval.
- Completed the catalog task and verified request closure.

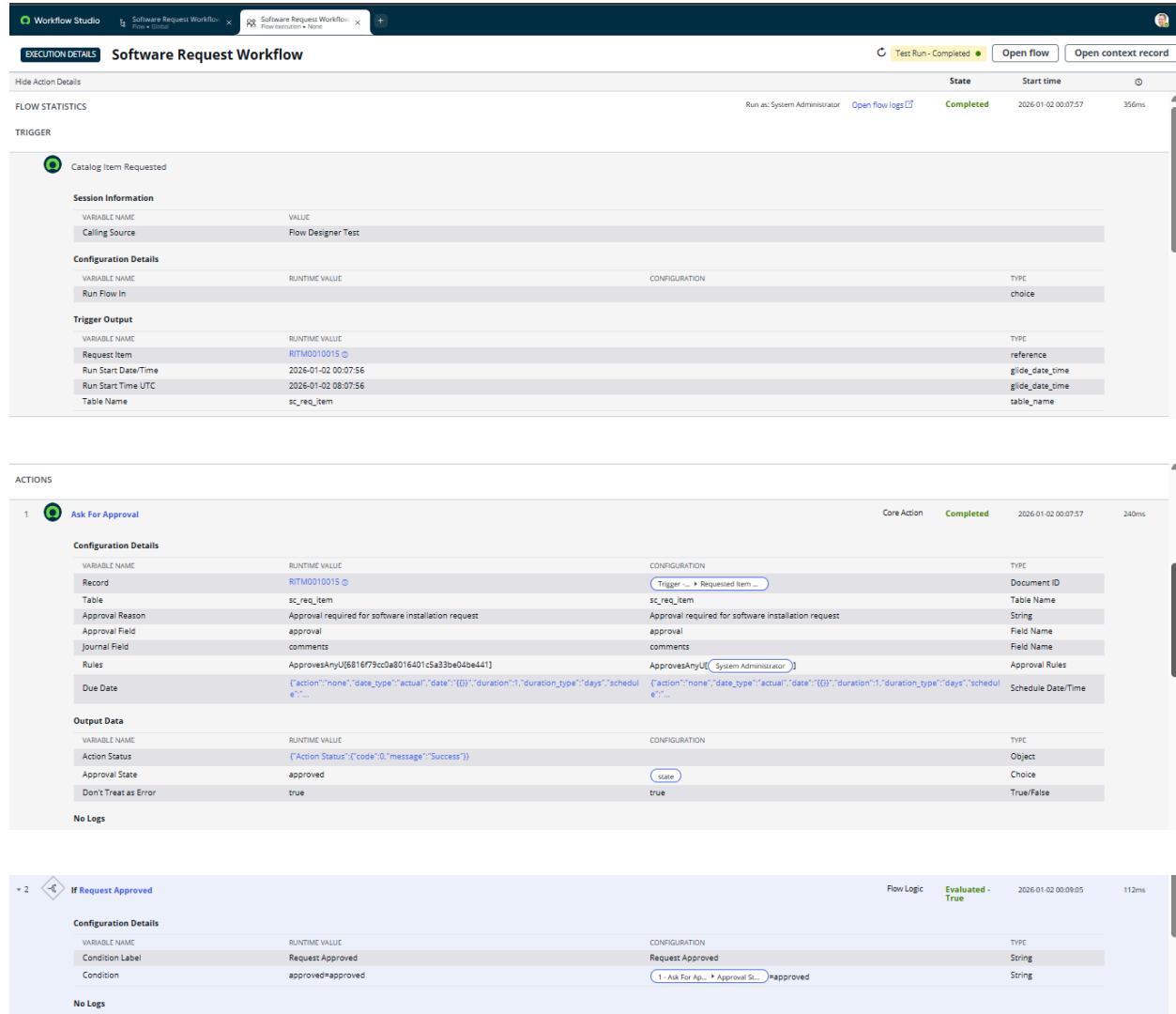
Observations

- Requests followed the correct lifecycle:
Submission → Approval → Catalog Task Creation → Fulfillment → Closure

- Flow execution logs reflected accurate status changes.
- Approval and task records were properly linked.

Screenshot Evidence (Optional)

Insert screenshot showing Flow Execution Details with successful completion.



Software Request Workflow

Execution Details

Session Information

VARIABLE NAME	VALUE
Calling Source	Flow Designer Test

Configuration Details

VARIABLE NAME	RUNTIME VALUE	CONFIGURATION	TYPE
Run Flow In	choice		

Trigger Output

VARIABLE NAME	RUNTIME VALUE	TYPE
Request Item	RITM0010015	reference
Run Start Date/Time	2026-01-02 00:07:56	glide_date_time
Run Start Time UTC	2026-01-02 08:07:56	glide_date_time
Table Name	sc_req_item	table_name

ACTIONS

1 Ask For Approval

Configuration Details

VARIABLE NAME	RUNTIME VALUE	CONFIGURATION	TYPE
Record	RITM0010015	Trigger ... > Requested Item ...	Document ID
Table	sc_req_item	sc_req_item	Table Name
Approval Reason	Approval required for software installation request	Approval required for software installation request	String
Approval Field	approval	approval	Field Name
Journal Field	comments	comments	Field Name
Rules	ApprovesAnyU([816f79cc0a8016401c5a33be04be44])	ApprovesAnyU([System Administrator])	Approval Rules
Due Date	{"action": "none", "date_type": "actual", "date": "(0)", "duration": 1, "duration_type": "days", "schedule": "e7..."}...	ApprovesAnyU([System Administrator])	Schedule Date/Time

Output Data

VARIABLE NAME	RUNTIME VALUE	CONFIGURATION	TYPE
Action Status	{"Action Status": {"code": 0, "message": "Success"}}	state	Object
Approval State	approved	approved	Choice
Don't Treat as Error	true	true	True/False

No Logs

2 If Request Approved

Configuration Details

VARIABLE NAME	RUNTIME VALUE	CONFIGURATION	TYPE
Condition Label	Request Approved	Request Approved	String
Condition	approved=approved	1 - Ask For Ap... > Approval St... = approved	String

The screenshot shows a ServiceNow interface for a workflow step named "Create Catalog Task".

Configuration Details:

VARIABLE NAME	RUNTIME VALUE	CONFIGURATION	TYPE
Table Name	sc_task	sc_task	Table Name
Requested Item	RITM0010015	Trigger → Requested Item ...	Reference
Short Description	Software installation task	Software installation task	String
Fields	assignment_group=31f2f16fb3c63210a67095d0dead3c9 description=Install the requested software as per ...	assignment_group{"display": "Software Support", "value": "31f2f16fb3c63210a67095d0dead3c9", "sys_id": "..."}	Template Value
Wait	true	1	True/False
Template Catalog Item			Reference
Catalog Variables			Slush Bucket

Output Data:

VARIABLE NAME	RUNTIME VALUE	CONFIGURATION	TYPE
Action Status	{"Action Status": {"code": 0, "message": "Success"}}	task	Object
Catalog Task	SCTASK0010025	true	Document ID
Don't Treat as Error	true		True/False

No Logs

Flow Logic: Completed, 2026-01-02 00:09:05, 112ms

4 End

The screenshot shows the "Software Request Workflow" in Workflow Studio.

EXECUTION DETAILS: Software Request Workflow

Test Run - Completed (Green dot) | Open flow | Open context record

State Start time Duration

Run as: System Administrator | Open flow logs | Completed | 2026-01-02 00:07:57 | 356ms

FLOW STATISTICS:

Core Action: Completed | 2026-01-02 00:07:57 | 240ms

Flow Logic: Evaluated - True | 2026-01-02 00:09:05 | 112ms

Core Action: Completed | 2026-01-02 00:09:05 | 112ms

Flow Logic: Completed | 2026-01-02 00:11:11 | 0ms

Flow Logic: Not Run

Core Action: Not Run

Flow Logic: Not Run

TRIGGER: Catalog Item Requested

ACTIONS:

- Ask For Approval
- If Request Approved
- Create Catalog Task
- End
- Else
- Update Record
- End

ERROR HANDLER:

Activity 4: Data Reliability Across Tables

Objective

To ensure data consistency across backend ServiceNow tables.

Tables Verified

- **sc_request (Request – REQ)**
- **sc_req_item (Requested Item – RITM)**

- sc_task (Catalog Task – SCTASK)

Observations

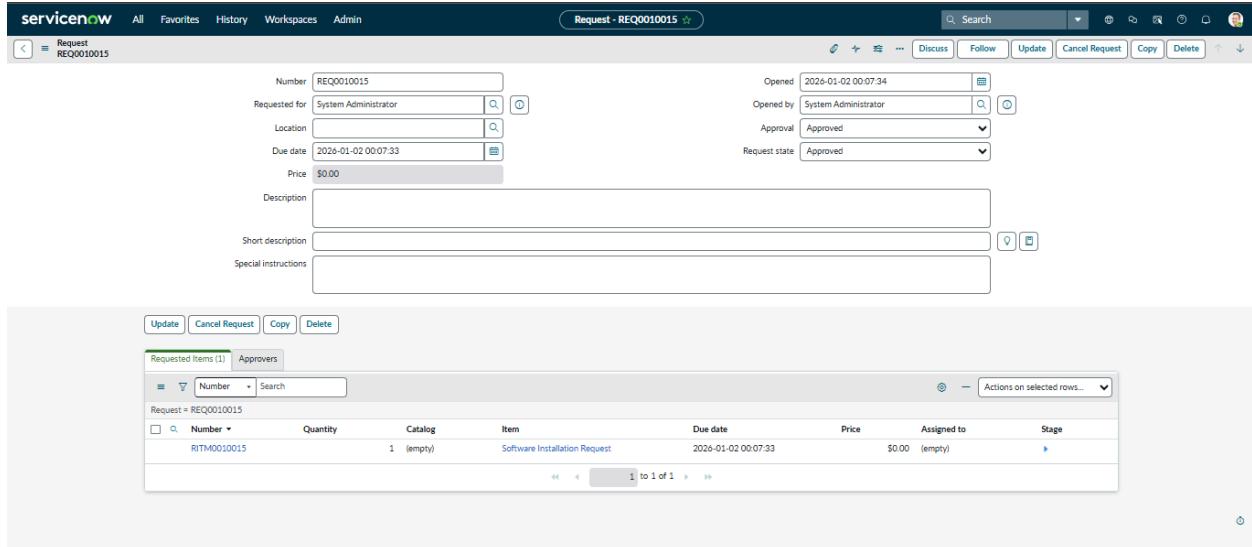
- All records were created automatically and linked correctly.
- Catalog variables were preserved and visible at the backend.
- Approval states were accurately reflected in RITM and REQ records.
- Task states transitioned correctly during fulfillment.

Outcome

These validations ensured strong data reliability, preventing unauthorized access, incomplete submissions, or inconsistent record states.

Screenshot Evidence (Optional)

Insert screenshot showing linked REQ → RITM → SCTASK records.



The screenshot displays the ServiceNow web interface for managing requests. At the top, the navigation bar includes links for All, Favorites, History, Workspaces, and Admin. The current view is titled "Request - REQ0010015". The main content area shows the details of a request record, including fields like Number (REQ0010015), Requested for (System Administrator), Location, Due date (2026-01-02 00:07:33), Price (\$0.00), Description, Short description, and Special instructions. Below the main record, a sub-panel titled "Requested Items (1)" shows a table with one item: RITM0010015, which is a Software Installation Request. The bottom of the screen features standard action buttons: Update, Cancel Request, Copy, and Delete.

Number	Quantity	Catalog	Item	Due date	Price	Assigned to	Stage
RITM0010015	1	(empty)	Software Installation Request	2026-01-02 00:07:33	\$0.00	(empty)	

servicenow All Favorites History Workspaces Admin Request - REQ0010015

Requested Item RITM0010015

Number	RITM0010015	Opened	2026-01-02 00:07:33
Item	Software Installation Request	Opened by	System Administrator
Request	REQ0010015	Stage	Request Approved
Requested for	System Administrator	State	Open
Due date	2026-01-02 00:07:33	Quantity	1
Configuration item		Estimated delivery	
Watch list		Backordered	<input type="checkbox"/>
		Order Guide	<input type="text"/>

Variables

* What software do you need?
intelliJ

Specify version (if required)
2021

* Why do you need this software?
Eclipse working space

* Select urgency level.
 Normal
 High
 Critical

Update Delete

Related Links Flow Context

Catalog Tasks (2) Approvers (2) Group approvals

for text Search Actions on selected rows... New

Request item	Number	Assignment group	Assigned to	Short description	Actual start	Actual end	Order
RITM0010015	SCTASK0010026	Software Support	(empty)	Software installation task	2026-01-02 00:09:04	(empty)	
	SCTASK0010025	Software Support	(empty)	Software installation task	2026-01-02 00:09:05	2026-01-02 00:11:11	

1 to 2 of 2

Catalog Task SCTASK0010025

Number SCTASK0010025 Assigned to Approval Not Yet Requested

Assigned to Software Support Priority 4 - Low

Assignment group Software Support State Closed Complete

Configuration item Request item RITM0010015

Active Requested for System Administrator

Short description Software installation task

Description Install the requested software as per approved request.

Work notes Work notes Post

Activities: 2

System Administrator State Closed Complete was Open Field changes • 2026-01-02 00:11:11

System Administrator Impact 3 - Low Opened by System Administrator Field changes • 2026-01-02 00:09:05

Opened by System Administrator Priority 4 - Low

Priority 4 - Low State Open

Update Delete

Affected Cls Approvers Group approvals

Confirmation team

Summary Table

Area	Actions Taken
Data Handling	Catalog variables mapped and validated across core ServiceNow tables
Access Control	Role-based approvals and conditional logic using Flow Designer
QA Testing	End-to-end testing including request, approval, task creation, and logs
Data Integrity	Mandatory validations, auto-fill logic, and approval state enforcement

Outcome

The Data Integrity phase successfully ensured that all Software Installation Requests maintained high data quality, security, and reliability. Mandatory validations, automated field population, and workflow-driven controls worked together to ensure accurate tracking and seamless fulfillment across all related ServiceNow records.