📄 Phase 5: Deployment, Documentation & Final Presentation

Document Technical Blueprint

Description

The Technical Blueprint documents the complete backend, functional, and automation design of the **Software Installation Request** solution. It provides a structured overview of how data is captured, processed, approved, fulfilled, and closed within ServiceNow. This blueprint serves as a technical reference for administrators, developers, and stakeholders to understand the system architecture, workflow behavior, and scalability considerations.

1️⃣ Data Model & Variables

Catalog Variables Implemented

The Service Catalog item captures the following key inputs from the user:

- **Software Name**
  Captures the name of the software requested for installation.
- **Version Required**
  Allows users to specify a particular software version if applicable.
- **License Justification / Purpose of Installation**
  Captures the business justification for the software request, ensuring compliance and audit readiness.
- **Urgency**
  Indicates the priority of the request (Normal / High / Critical) to support operational planning.
- **Requested For**
  Auto-populated with the logged-in user but editable to allow requests on behalf of other users.

All variables are stored using standard ServiceNow Service Catalog tables and are linked to the **Requested Item (RITM)** for backend processing and fulfillment.

2️⃣ Approval Architecture

Approval Use Cases

The solution supports structured and conditional approvals using **Flow Designer**:

- **Manager / Designated Approver Approval**
  All software installation requests are routed for approval before fulfillment.

- **Conditional Approval Logic (Design-Ready)**
  The workflow supports extension for:

  - **Software Asset Team approval** when licensed software is requested.

  - **IT Security approval** for high-risk software or elevated access requirements.

Approval records are stored in the **sysapproval_approver** table and linked to the RITM, ensuring proper audit trails and approval visibility.

3️⃣ Workflow & Automation Logic

Flow Designer Workflow Overview

The automation is implemented using **Flow Designer**, following this execution sequence:

1. Service Catalog submission by the user

2. Automatic creation of Request (REQ) and Requested Item (RITM)

3. Manager approval request

4. Conditional approvals (if applicable)

5. Catalog task creation for IT fulfillment
6. Workflow waits for task completion
7. Automatic closure of RITM after successful fulfillment

Rejection Handling

If the request is rejected:

- The RITM is updated to reflect rejection
- A backend **Business Rule** triggers automatic **Incident creation** for tracking and escalation

This ensures controlled execution and prevents unauthorized or incomplete software installations.

4️⃣ Fulfillment Task Design

Fulfillment Task Use Cases

- **Catalog Task Creation**
  Tasks are generated automatically for the **IT Software Support group** upon approval.
- **Task Lifecycle Management**
  Tasks follow standard ServiceNow states:
  - Open
  - Work in Progress
  - Closed Complete
- **Automatic RITM Closure**
  The Requested Item is closed automatically once all associated catalog tasks are completed successfully.

This design ensures accurate fulfillment tracking and operational accountability.

5️⃣Variable-to-Field Mapping Logic

Catalog variables captured during request submission are logically mapped into fulfillment records:

- **Software Name** → Catalog Task short description
- **Version Required** → Task description
- **License Justification** → Approval and audit context
- **Urgency** → Task priority (logical handling)

This mapping ensures data consistency across request, approval, and fulfillment stages.

6️⃣Exception & Incident Handling

To handle exception scenarios:

- A **Business Rule (build script)** is implemented to automatically create an **Incident** when:
    - A request is rejected
    - License availability or compliance issues are detected

This provides:

- Proper escalation
- Audit compliance
- Visibility into blocked or failed requests

7️⃣Core Tables Involved

The following ServiceNow tables form the backbone of the solution:

- **sc_cat_item** – Service Catalog Item definition
- **sc_request (REQ)** – Service Catalog Request
- **sc_req_item (RITM)** – Requested Item
- **sc_task (SCTASK)** – Fulfillment tasks
- **sysapproval_approver** – Approval records
- **incident** – Exception handling and escalation

8️⃣Custom Data Handling & Scalability

Although no custom tables were created in the current implementation, the solution design supports future extensions such as:

- License category tracking
- Cost center or budget mapping
- Software asset metadata

This ensures the solution is scalable without redesigning the workflow.

9️⃣Portal Customization References

No custom Service Portal widgets were developed for this project.
The solution uses the standard ServiceNow Service Portal for request submission and tracking, ensuring maintainability and platform compatibility.

Outcome

The Technical Blueprint provides a comprehensive and unified view of the solution architecture. It clearly documents data flow, approval logic, automation behavior, fulfillment handling, and exception management. The design is transparent, scalable, maintainable, and fully aligned with ServiceNow best practices.