# Task 20: Requirements Discussion – Service Catalog

*Translating Services into Clear, Actionable Request Forms*

The Service Catalog is not just a feature—it is the face of your service organization. It defines what users can ask for, how requests are routed, and who fulfills them. Properly implemented, it drives clarity, accountability, and efficiency.

This task is rooted in the ITIL practices of **Service Catalog Management**, **Request Fulfillment**, and **Service Design**. It also directly influences downstream automation, approvals, and performance metrics.

What It Is

This phase focuses on collecting, refining, and organizing the service request items that will be visible to end users. It includes determining:

* The list of catalog items (typically capped at 25 to start)
* Field requirements for each item (text, dropdowns, attachments, etc.)
* Visibility rules (who sees what items)
* Required approvals and fulfillment workflows
* Logical groupings into **Service Categories** (up to ten commonly used)

The output of this discussion directly feeds into configuration work, workflow automation design, and portal setup.

Why It Matters

A well-structured catalog reduces ticket clutter, speeds up request fulfillment, and minimizes back-and-forth with requesters. It also encourages self-service adoption, especially when items are intuitively grouped and clearly worded.

ITIL’s **Service Catalog Management** emphasizes maintaining a current, accurate, and user-friendly view of all operational services. This ensures that requesters only see services for which they are eligible and that support teams are not overwhelmed by misrouted tickets.

A catalog also reinforces standardization, which is crucial for maintaining Service Level Agreements (SLAs), tracking demand, and enabling future automation.

**How To Do It**

1. **Gather Candidate Service Items**
   * Start with a stakeholder-supplied list of services (up to 25 recommended).
   * Ask each business unit (IT, HR, Facilities) to define what they regularly provide to users.
   * Identify any legacy forms or workflows currently in use.
2. **Define Field Requirements**
   * For each item, document the field names, field types, and conditional logic.
   * Include custom fields where needed (e.g., “Asset Tag #” or “Manager Email”).
   * Identify which fields are required and which are optional.
3. **Design for Visibility**
   * Use **group visibility rules** to restrict access based on department, role, or location.
   * Consider who should be able to view, submit, or fulfill each request.
4. **Determine Approval Workflows**
   * Specify if a catalog item needs approval.
   * Document approval logic: e.g., “If cost > $500, send to Department Head,” or “Always send to manager using manager lookup.”
5. **Categorize for Structure**
   * Group service items into **no more than 10 Service Categories** for launch (e.g., “Access Requests,” “Hardware,” “HR Services”).
   * Use clear, user-friendly labels.
   * Decide if catalog items need subcategories (optional).
6. **Automation Planning**
   * Identify catalog items that will need automation.
   * Define up to 8 initial **Workflow Automators** tied to catalog items (e.g., route a “New Laptop Request” to IT Procurement and auto-assign a task to Device Prep).
   * Tag which workflows will include SLA escalation, task generation, or notifications.

Tips for Success

* **Think like the user**: Use plain language in service item names (“Request Email Access” instead of “Mailbox Provisioning Form”).
* **Don’t overload the launch**: Limit the first wave to 25–30 items. You can always add more later.
* **Standardize patterns**: If several forms ask for the same fields (e.g., location, department), use consistent naming and formatting.
* **Include logic early**: Plan out workflow branches, approval conditions, and notifications now to avoid rework during configuration.

**ITIL Tie-In**

This phase activates multiple ITIL practices:

* **Service Catalog Management**: You are creating and curate the services visible to users.
* **Request Fulfillment**: Each service item becomes a repeatable, traceable workflow with defined inputs and outputs.
* **Access Management**: Many service items request access to systems—requiring controlled logic and auditability.
* **Service Level Management**: Each catalog item can carry an SLA for response and resolution—plan these accordingly.
* **Change Enablement**: If a service item triggers provisioning or access that requires change control, define the escalation path now.

This Task also reinforces the ITIL guiding principle: *Focus on Value*. Every item in the Service Catalog should exist to solve a user problem or provide a repeatable service that brings business value.

When thoughtfully designed, the Service Catalog becomes more than a form repository—it becomes the most effective entry point for service delivery. Your goal is to make it functional, flexible, and approachable. Build it with the requester and the agent in mind, and it will serve both equally well.