

itSMF International
The IT Service Management Forum

IT Service Management GLOBAL BEST PRACTICES

Collector's Edition

7.1 Introduction

With the introduction of ITIL® V3, the primary focus of ITIL shifted from a process approach to a service lifecycle approach. However, ITIL V3 has not abandoned the process approach with which we are all familiar. All ITIL V2 processes are still present in one or more of the service lifecycle phases, and many processes, functions and other activities have been added to ITIL.

Further, a lot of organizations still seem to live by the process approach. A lifecycle approach, as suggested in ITIL V3, assumes a certain level of maturity of both the business (demand) and IT service provider (supply). Chances are that many organizations are still busy getting the core support and delivery processes up and running. This means that assistance in designing and implementing those processes remains relevant, and this is what this chapter is all about.

That doesn't mean that new ITIL concepts go unmentioned; on the contrary: the Configuration Management System (CMS) and the Service Knowledge Management System (SKMS) are explained in detail, and a major part of this chapter is dedicated to three processes that have earned a place of their own in ITIL V3: service portfolio management, service catalog management and request fulfillment.

Three articles each cover one of these processes, and together they give an excellent overview on the subject and how they are related.

At the end of this chapter there is guidance on an old, but challenging issue: the relationship between change and release management and project management.

CONTENTS

This chapter includes six articles about designing and implementing ITIL processes:

- **Integrating configuration management into existing processes**
Author: Chiu-Ping Kuo (Acer Inc., Taiwan)
- **Bringing wisdom to ITSM with the Service Knowledge Management System**
Authors: Bryce Dunn and Linh C. Ho (Compuware USA)
- **Starting right with the service portfolio and service catalog in the information age**
Author: Mark O'Loughlin (IT Alliance, Ireland)
- **Service catalog management - Best practices and practical advice**
Authors: Bill Fine and Nick Schneider (newScale, Inc. USA)
- **Request fulfillment - Ten best practices for managing IT service requests**
Author: Rodrigo Fernando Flores (newScale, Inc. USA)
- **The unclear relationship between change, release and project management**
Authors: Christian Cantù and Maxime Sottini (iCONS - Innovative Consulting S.r.l., Italy)

SHORT SUMMARIES

Configuration management is one of the most complicated and time-consuming processes to design and implement. For practitioners new to this field, there is always a gap between theory and practice. The article by **Chiu-Ping Kuo** assists these practitioners in reducing that gap. Her article consists of three parts. Firstly, the design principles for successful configuration management are provided. Secondly, a step-by-step guide is given on how to design the CMDB, develop the configuration control process, and assign responsibilities to individuals. Each step is coupled with methods and techniques. Finally, she lists the benefits of configuration management for other ITIL processes with a CMDB perspective.

The complaint that there is “too much data but not enough information” is not uncommon. Providing the right information to the right people at the right time is a difficult challenge to face. But with effective knowledge management in place, both the IT service provider and customer can benefit significantly. Although knowledge management is a mature discipline of its own, it has never received much attention in ITSM until the publication of ITIL V3. In their article, **Dunn and Ho** present the what, why, how and who of the Service Knowledge Management System (SKMS), which plays a central role in ITIL’s knowledge management process.

One of the most common mistakes that an organization can make is to mix up the concepts of service portfolio and service catalog. A lot of organizations start developing what they perceive to be a service catalog when, in fact, they are developing a service portfolio. **O’Loughlin** explains the differences between the two, and provides guidance on the basics of building a service portfolio framework. He then continues on how building a service portfolio will aid in the development and implementation of a service catalog. His article closes with process descriptions of two major aspects of service portfolio management.

The service catalog is an essential element for a successful, service-centric approach to IT service management. It can be used by IT to manage the service lifecycle, to market available services to the business, and it can be used as a means to manage day-to-day service requests. But how do you create an effective service catalog? In their article, **Fine and Schneider** show you how it’s done. They outline a step-by-step approach to designing and rolling out an IT service catalog and use real-world examples.

After the articles on the service portfolio and service catalog, it’s time to discuss the final piece of the puzzle: the request fulfillment process.

The current increase in the volume and complexity of service requests, can be too much for an IT service provider who does not have an efficient request fulfillment process in place. In his article, **Flores** shows how to get request fulfillment under control. He outlines ten guidelines that help an IT organization to provide more responsive service delivery, increase operational efficiency, and improve customer satisfaction.

The relationship between change, release and project management is not well addressed by two of the existing best practice frameworks, ITIL® and PRINCE2™. In their article, “The unclear relationship between change, release and project management”, **Cantù and Sottini** unveil the relationships between the processes in detail. They do so by categorizing them in specific sections: they explore the relationships between change and release management, change and project management, and project and release management.

7.2 Integrating configuration management into existing processes

To implement configuration management in a complex environment, organizations must apply a systematic approach to the design of the CMDB, develop integrated control processes, and assign appropriate roles and responsibilities. Based on a case study, Chiu-Ping Kuo explains all the ins and outs of a configuration management implementation.

INTRODUCTION

The efficiency and effectiveness of ITIL® processes determine the success of an ITIL implementation. Based on the adoption of ITIL processes, many organizations claim they are ITIL-compliant. However, on gaining a deeper insight into their ITIL processes, it could turn out to be nothing more than a lot of activities and paperwork. Either the collected service operation data is not sufficient, or the collected data is not transformed into meaningful information. In order to exemplify the value of ITIL processes, organizations need to efficiently control and manage operation data related to their IT infrastructure and service operations. Configuration management is a key step in ITIL implementation to achieve this purpose. Process integration is a great challenge due to the databases and procedures already in place in most organizations. To implement configuration management in a complex environment, organizations can apply a systematic analysis and design approach to the design of the Configuration Management Database (CMDB), development of integrated control processes, and assignment of related roles in the analysis and design phases.

Overview

In this article, a real case example is introduced to convey the practical experience of an organization in implementing configuration management. The organization, like many others, had a general IT environment and procedures in place before it implemented the process. Based on the case, this article consists of three parts. Firstly, the design principles to key elements of successful configuration management are provided as a basis for the ensuing design work. Secondly, a step-by-step guide is provided on how to design the CMDB, develop the configuration control process, and assign responsibilities to individuals. Each step is coupled with methods and techniques for practitioners to overcome possible obstacles in this stage. Finally, from the perspective of the CMDB, the benefits for other ITIL processes are summarized.

Purpose

Configuration management is one of the most complicated and time-consuming ITIL processes to design and implement. For practitioners new to this field, there is always a gap between theory and practice. This article is intended to diminish the gap through the presentation of a practical design together with the associated outputs. With the knowledge provided in this article, readers will be better able to capture the essence of configuration management and make it work in practice.

This article focuses on both analysis and design. The target audience includes process practitioners, CMDB designers, tool implementers, and operation managers.

EDC CASE DESCRIPTION

The E-business Data Center (EDC)

The E-business Data Center (EDC) is a business unit of an international IT company that supplies products and services. EDC delivers IT infrastructure management and outsourcing services to its customers. Comprising state-of-the-art facilities and IT services, EDC optimizes business operations to enable customers to compete successfully with reduced risks and costs on its IT infrastructure.

EDC provides services including dedicated and shared command center service for disaster recovery, tape depository and backup service, co-location service, network and system monitoring service, and security monitoring service.

EDC has two data center facility locations. The primary location is 25 miles south of the company's headquarters. An alternate site is operated as a satellite hosting facility near the company's headquarters.

There are approximately 200+ employees in EDC, which runs as a profit center with its own business and operational departments. This international IT company sets up supporting functional units in the head office to perform centralized management, such as human resources management, finance management, legal management, etc. EDC is organized as shown in figure 1.

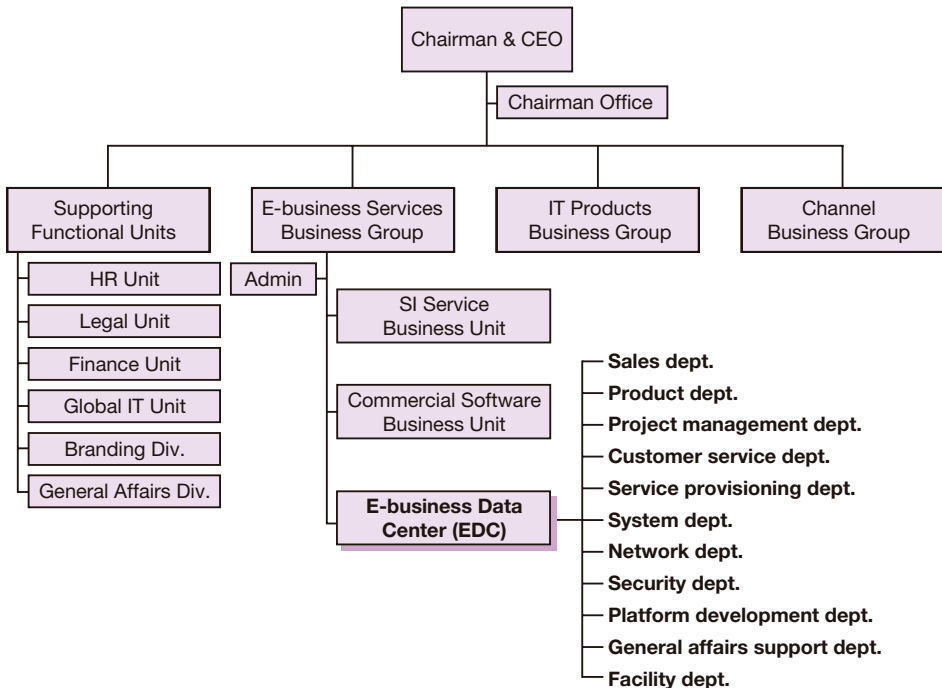


Figure 1 Organization chart of EDC

Situation before implementing configuration management

EDC started implementing configuration management six years after its establishment in 2000. At that time, most IT systems had been set up and many operational processes had been developed to support its business. The following sections briefly describe EDC's situation before this CMDB initiative.

IT environment

Software/Application

1. HR management system (HRMS) - HRMS is used by the HR unit to handle payroll, employee evaluation, training and other related affairs. The system is developed in-house and owned by the head office.
2. Financial management system (FMS) - FMS provides functions for planning/budgeting, internal controls evaluation and monitoring. Asset management is a FMS component that supports the procurement process. Contract management is another essential component of this system. The supporting functional units use FMS to carry out their functions.
3. Project management system (PMS) - E-business Services Business Group (EBG), which EDC is a part of, uses PMS to carry out central management of project planning and control. The users of the system include all business units under EBG.
4. Service process management system (SPMS) - SPMS is a customized management system based on a commercial ITIL compliance tool. SPMS provides a platform for managing service processes including incident and problem management processes. A web-based service portal is part of the system. A platform development department is responsible for the customization and maintenance of this system.
5. Document management system (DMS) - DMS, an outsource-developed system, is owned and used by EDC to manage documents, including version control, document release, approval, etc.
6. Other commercial products - There are many other commercial products used to support EDC businesses. These products are either EDC's or head office's assets:
 - assets belonging to EDC include network monitoring and management tools, system monitoring and management tools, security event monitoring tool, tape backup tools, building management system, version control system for source code, and database management system
 - assets belonging to head office include email system, operating systems, and personal computer related software

Hardware

To support the IT operations, EDC has a cluster of 20 servers at the primary location and 6 servers at the alternate site. These servers are primarily Windows, Linux, or AIX machines for delivering EDC services. Additional servers are located at head office to provide internal management services. Two tape libraries are used to support the tape backup service. The network environment consists of 2 T3 lines, 2 Fast Ethernet lines, and other necessary network equipments. Each employee has his/her own personal computer or laptop.

Existing processes

Two types of processes exist at EDC. One is at the business unit level, which is developed by EDC and is applicable to EDC only. Another is at the corporate level, which is applicable to the whole company and, in the main, has been operating for over 20 years.

Business unit level

The incident, problem and change management processes have been implemented with SPMS. Service level management has not been implemented completely but the service catalog is available to customers. The document release process is established with DMS. Application development was also established.

Corporate level

Processes such as procurement, human resources management, asset management, contract management, and project management are developed and maintained by the head office.

Challenges in the planning stage

Like most other organizations, EDC faced challenges at the outset from the following issues:

1. Data is stored in different locations with existing tools. It poses challenges to the idea of the central management of information.
2. The right CMDB scope and Configuration Item (CI) level are subjectively understood by individual employees without any consistent foundation.
3. Many processes are already well-defined and operated. Leveraging existing processes is a necessary requirement.

KEY ELEMENTS OF SUCCESSFUL CONFIGURATION MANAGEMENT

An appropriate CMDB, integrated control processes, and effective audit activities all contribute to the success of configuration management. These three elements are designed in sequence. In the following sections, several general design principles used in the EDC case will be presented.

An appropriate CMDB

The CMDB is a tool to link business services to IT infrastructure. Under this concept, the CMDB should provide functions to support business and operational activities, for example, investment decision making, impact analysis, service quality control, among others. It is a significant challenge to specify the appropriate CMDB scope and a proper CI level in the planning stage. Many practitioners only realize their CMDB is not well-designed after encountering difficulties in supporting certain processes during operation. It is a complicated task to restructure the CMDB, especially when associated tools have been implemented and the CMDB has been populated. To avoid duplicated efforts, the following principles are useful in CMDB design:

1. Use both a “top-down” and a “bottom-up” approach.
2. Use “service” as the starting point.
3. CIs and their attributes should adhere to business service operation.

Integrated control processes

Configuration management is responsible for providing an accurate CMDB to support other ITIL processes. To get the CMDB into smooth operation, a precisely controlled process is needed throughout the lifecycle of a CI. In a complex IT environment, a CMDB normally contains multiple data sources. Before planning the CI control process, it is necessary to carefully analyze all data sources and their related procedures. Some critical design principles are listed below:

1. Design the CI control process with the change process.
2. Clearly define the data source and keep data in its original location as far as possible.

3. CI status should be traceable.
4. Leverage existing working processes where appropriate.

Effective audit activities

The purpose of a configuration audit is to ensure reliable information in a CMDB. For an effective audit activity, three aspects have to be considered when developing an audit plan:

1. Verify the CI information in terms of integrity and consistency.
2. Check the completeness of the CI update process.
3. Auditors should be familiar with the specific CI categories.

METHODS AND TECHNIQUES

CMDB design

EDC took three steps to accomplish the CMDB design work. The first step was to define the requirements of the CMDB. The second step was to gather service relevant information and analyze it. Finally, the CMDB model with selected CI instances would be developed. Methods and techniques used by EDC for each of these steps are introduced in the following sections.

1. Defining the requirements of the CMDB

A CMDB is at the core of configuration management. To design a CMDB, a policy on what data is to be included in the scope should be developed first. In the EDC case, the CMDB requirements are defined in a configuration management policy. They cover the following contents:

1. All EDC businesses should be included. Non-EDC businesses should be outside of the scope.
2. Components of each EDC business are considered as CIs only if they are both critical and “EDC-owned”.
3. Information on customer contracts, OLAs or UCs should be part of the CMDB.
4. The CI details should contain CI characteristics, capability information, control information, and status accounting information.
5. The lowest CI level is defined as follows:
 - Purchase unit is the lowest level for physical CIs, such as hardware, software, licenses, etc.
 - Application level is the lowest level for in-house developed applications.
 - Contract level is the lowest level for customer contracts, OLAs, and UCs.

2. Analysis of service infrastructure

Business services and IT infrastructure are the major targets for analysis. A “top-down” approach is applicable to the business service analysis, from which the business portion of the CMDB will be depicted. Similarly, a “bottom-up” approach is adopted to perform the IT infrastructure analysis, from which the physical portion of the CMDB will be depicted. EDC used a service infrastructure analysis form, of which an example can be found in table 1, to gather the high level information on the service infrastructure. In the service infrastructure analysis form, data is grouped into six types (“data groups”) - business service, software/AP, server, network, storage, and peripheral. When using this analysis form, EDC followed the following rules:

1. Based on the configuration management policy described in the previous section, for the data group “business service”, only EDC business services are recorded in the “Item” column of the analysis form.

2. Each item gathered will be placed in one data group.
3. The relevant information for each item, such as ownership and location, must be recorded.
4. After gathering all information, there is a need to determine the necessity for each item to the business services. An analysis result of “Critical component” or “Information required only” will exist at the “Required to service” column of this analysis form.

The gathered information in the columns of “Ownership” and “Required to service” will be used in the follow-up step.

Data group	Item	Ownership	L	R	Comments
Business services	Dedicated command center	EDC	-	-	
	Tape backup	EDC	-	-	
	System monitoring	EDC	-	-	
	Network monitoring	EDC	-	-	
	[more]				
Software/AP	HRMS	Head office	H	I	Shared service
	FMS	Head office	H	I	Shared service
	PMS	EBG	H	I	Shared service
	Network monitoring tool	EDC	S1	C	Network monitoring
	System monitoring tool	EDC	S1	C	System monitoring
	Building management system	EDC	S1	I	Shared service
	Email system	Head office	H	C	Shared service
[more]					
Server	Network monitoring server	EDC	S1	C	Network monitoring
	System monitoring server	EDC	S1	C	System monitoring
	Tape backup server -1	EDC	S1	C	Tape net backup
	Tape backup server -2	EDC	S2	C	Tape net backup
	FMS server	Head office	H	I	Shared service
	[more]				
Network	Core Router -1	EDC	S1	C	IT infrastructure
	Core Router -2	EDC	S2	C	IT infrastructure
	T3 -1	Supplier A	S1	C	IT infrastructure
	Ethernet -1	Supplier B	S2	C	IT infrastructure
	[more]				
Storage	Tape library -1	EDC	S1	C	Tape backup service
	Tape library -2	EDC	S2	C	Tape backup service
	[more]				
Peripheral	Printer -1	EDC	S1	C	Command center
	Fax -1	EDC	S1	C	Command center
	[more]				
Location(L):head office(H), EDC's prime location(S1), EDC's secondary location(S2)					
Required to service(R): information required(I), critical component(C)					

Table 1 Example of a service infrastructure analysis form

3. CI identification and CMDB model

The result of the service infrastructure analysis form has highlighted the potential CI data. The next step is to identify the precise CI data in the CMDB scope and develop the CMDB model. Copyright protected. Use is for Single Users only via a VHP Approved License. For information and printed versions please see www.vanharen.net

It is important not to include all gathered data from the previous step into the CMDB. Several activities need to be carried out in this step:

1. Perform the analysis of each business service.
2. Design the data structure for each business service.
3. Design the CMDB model and define CI attributes.

Each activity is described in detail below.

1. Performing the analysis of each business service

The purpose of this activity is to decompose a business service into components which could be further identified as CIs in the subsequent activity. The decomposition level is to the lowest CI level, defined in the configuration management policy. EDC utilized a well known analysis tool, Ishikawa Diagram, to analyze factors supporting its business services. The factor categories in the diagram are derived from the data group in the service infrastructure analysis form. EDC defined 8 factor categories to perform the analysis, these being: software/AP, server, network, storage, peripheral, service content, people, and document. The system monitoring service of EDC is used here to demonstrate the analysis result. See figure 2.

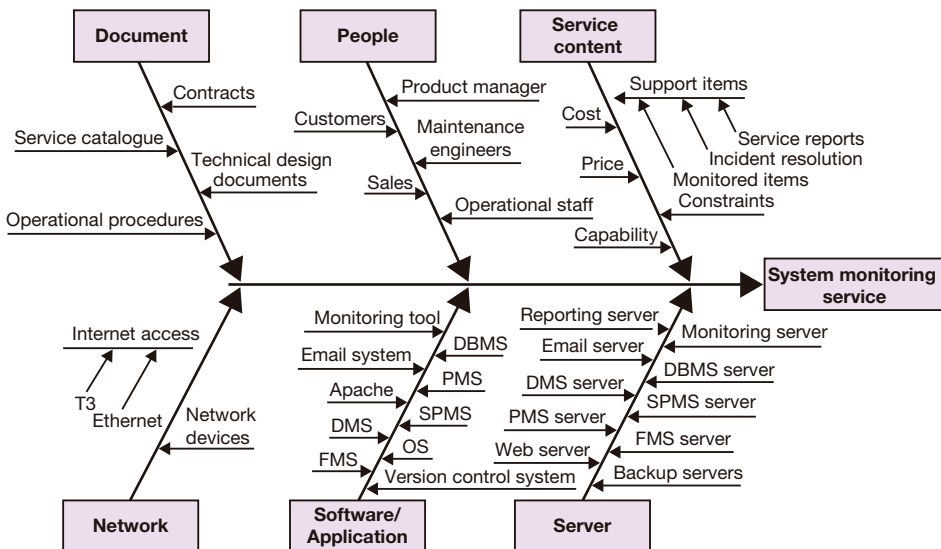


Figure 2 Example of analysis result

2. Designing the data structure for each business service

The next activity is to convert the analysis results into a data structure diagram, to which CIs are determined and CI relationships are established further. Taking the system monitoring service as an example, the resultant data structure is shown in figure 3. Items listed in each block of the diagram represent potential instances of CI, with the exception of the “System Monitoring Service” block. The potential CI instances are confirmed to be in the CMDB scope by a mix of the business service analysis result and the service infrastructure analysis result (see the section “Analysis of service infrastructure”). If the potential CI instances are in the categories of software/AP, server, network, storage, or peripheral, they need to meet the criteria of “Critical component” and “EDC-owned”. For example, EDC needed

the email function to deliver a business service. However, the head office of the company is responsible for the maintenance of the email server and system. Thus, both the email server item and the email system item are not identified as CIs. Instead, an Operational Level Agreement (OLA) between EDC and the head office is put in place to ensure the email functionality. This way it is possible to make sure all identified CIs are correct.

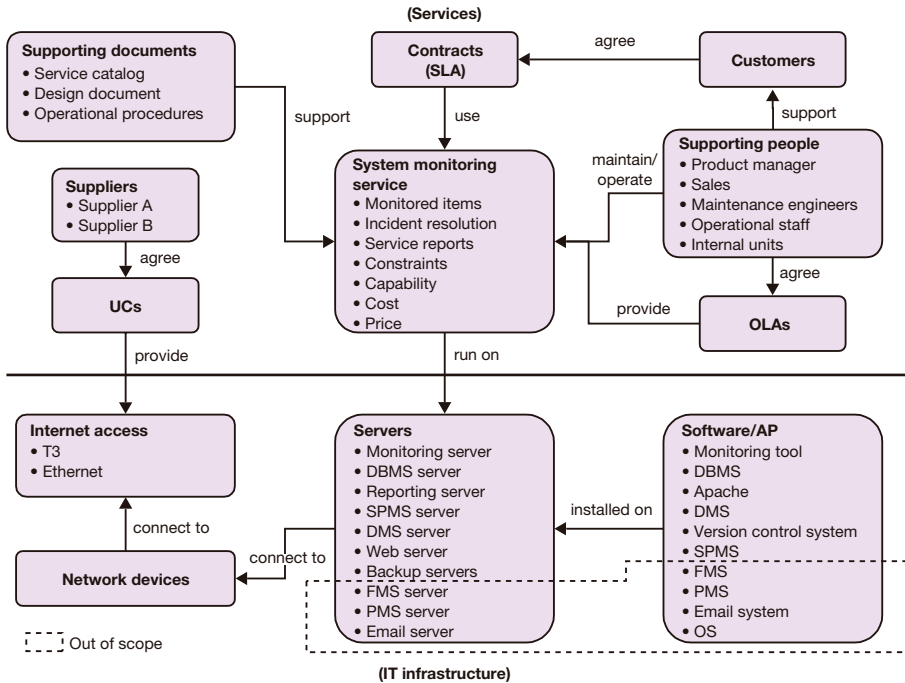


Figure 3 Example of a data structure

3. Designing the CMDB model and defining CI attributes

After completing the analysis of each business service, a union task has to be performed, which forms the final CMDB model, also called a CMDB schema. See figure 4 for the final CMDB model of the EDC case.

The final activity is to define the attributes of each CI category, which is determined by the information sufficiency requirements for business service operation. There are four areas of information to be considered when defining attributes:

1. CI characteristics, such as unique identifier, name, type, version, components, cost, price, etc.
2. Capability information, such as specification, functionality, performance/resource threshold, service hour, etc.
3. Control information, such as ownership, contact information, access right, tracking information, etc.
4. Status accounting information, such as lifecycle state, create date, expired/end date, change performer, location, etc.

It is recommended that a regular review of the CI attributes should be carried out to assess their sufficiency in service operation.

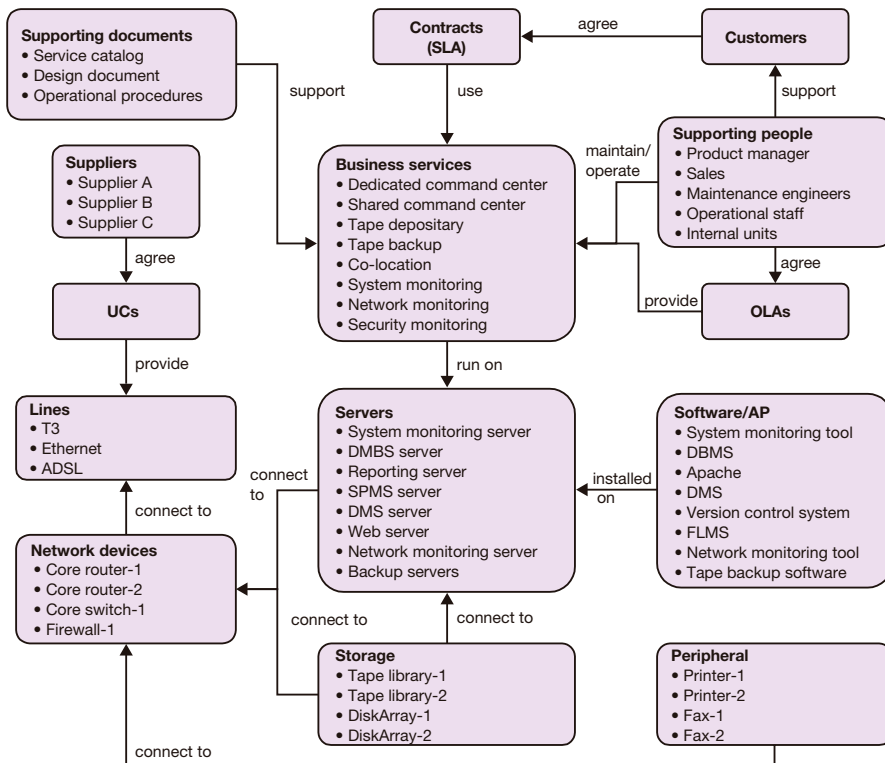


Figure 4 The CMDB model

Interfaces with other processes

The CMDB model is the logical framework in which the controlled data scope and relationship are clearly depicted. The follow-up challenge is to make the logical model work in practice. EDC adopted a 3-steps method for this task.

Designing the CMS architecture

The Configuration Management System (CMS) is a tool used by configuration management to support all ITIL service management processes. The CMS architecture design is an important task for building a robust supporting basis since the elements of the system and their relationships will influence the way it supports the management process. The CMDB approach and the system elements determine the CMS architecture. The practical methods are described below.

Choosing a CMDB approach

Without any doubt, the CMDB should be one element of the CMS. A correct CMDB approach would further affect the process integration and collaboration. In general, there are three types of CMDB approach, each adopted for different reasons:

1. The “Centralized approach” creates a central CMDB containing all CI details. The CI data is directly stored in the CMDB instead of in its native stores. Even though this approach involves costly data migration and changes to existing workflow, some small organizations adopt this approach for simpler database management and CI data access.

2. A “Federated approach” where data sources remain in their respective data stores, but a central CMDB is created to hold key CI data with links to CMDB extended data, such as incidents, RFCs, contracts, SLAs and so on. Data is imported into the central database from other data stores. This approach could enjoy the advantage of reducing the impact to existing procedures.
3. A “Virtual/distributed approach” relies on disparate databases distributed across the IT environment. Without a central CMDB, data access is directly to the respective application databases. From the perspective of database and workflow, this seemingly simple and fast means of implementing a CMDB creates the complication and difficulty of information presentation.

In the EDC case, there were two key factors influencing the decision making as far as the CMDB approach was concerned. Firstly, since EDC has been operating for a while, the CI data was stored in a mix of inventory and other specialized tools. Secondly, as a business unit in an international company, EDC was not allowed to change any corporate level processes. Thus, EDC decided to take the federated CMDB approach, whereby all application databases and corporate level processes remained unchanged. The additional requirements of the federated CMDB were the creation of a central database and a data collection mechanism which performs data synchronization, data extract and load, or data transformation techniques. Figure 5 depicts the federated data model chosen by EDC.

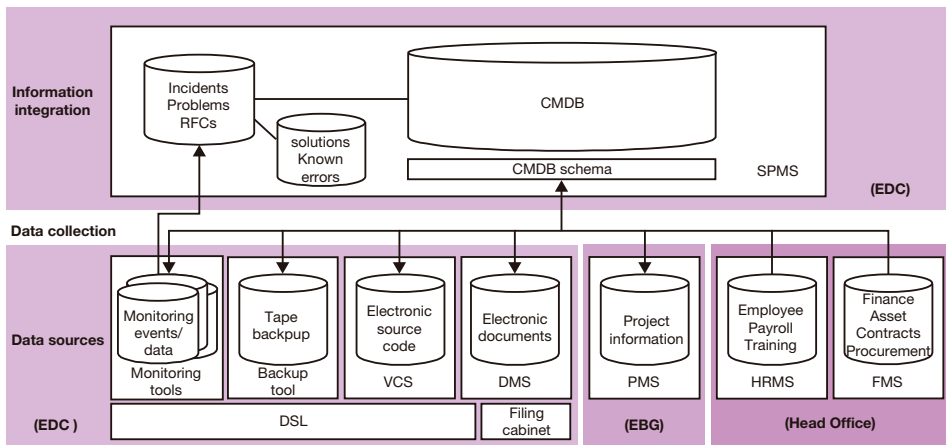


Figure 5 The federated CMDB of EDC

Defining the architecture elements

When designing the CMS architecture, People, Process, and Product/Tool (3 Ps) can provide a valuable input. As information is ultimately generated for people, the information presentation for different stakeholders will be a necessary element in the CMS. In addition, there should be a data processing mechanism between the information presentation and the CMDB that performs the task of information generation. Normally, the data processing mechanism will combine working processes and automation tools to ensure people receive accurate and complete information. Based on the 3 Ps and the chosen CMDB approach, EDC defined its CMS architecture elements as 4 layers:

1. The information and data layer (the federated CMDB) which includes the CMDB- and service operation-related information.

2. The tool layer, consisting of applications, management tools, and the IT infrastructure, which store data or information in appropriate repositories and operate under the guidance of working processes.
3. The process layer which includes working processes and procedures. Each process or procedure defines a set of activities to process specific information. This layer relies on tools to ensure that it meets its efficiency requirements.
4. The information presentation layer relies on user interfaces or reports to represent information to users, operation staffs, managers, or customers. People depend on this information to make decisions or take actions.

Together the tool layer and the process layer perform activities such as query, impact analysis, status accounting, monitoring, and so on. The outputs of these activities are associated with certain values or functions. Therefore, they can be treated as a knowledge processing layer. The simplified CMS of EDC is show in figure 6.

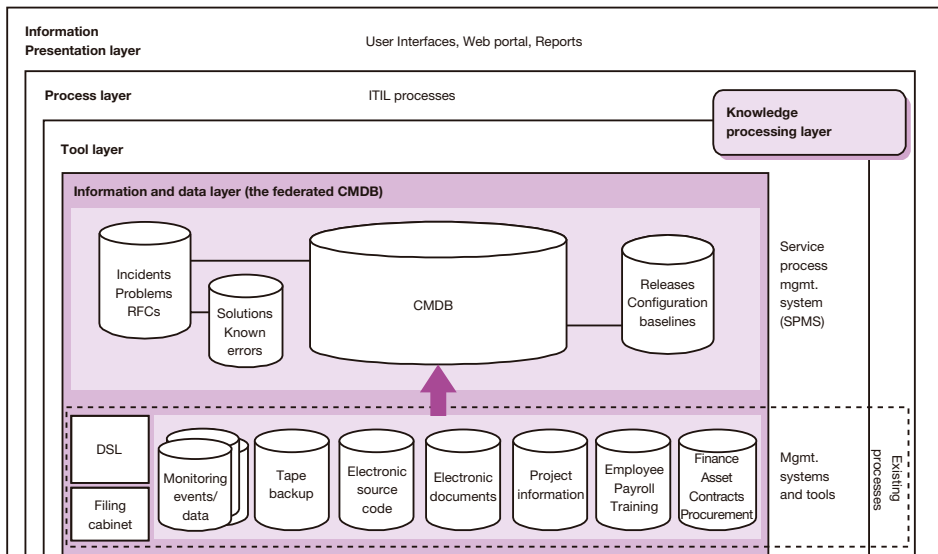


Figure 6 Example of a CMS architecture

Analyzing CIs' relationship to process

In order to leverage existing working processes, it is essential to analyze the relationship between CI data and existing processes. The CRUD matrix is a technique to map data to processes. CRUD stands for Create, Read, Update and Delete. Processes are placed on one axis of the matrix and data on the other axis. A letter C, R, U, or D is shown at the intersection between one row and one column if there is a relationship. EDC used this technique to undertake the analysis from which the CI control process was developed further. Based on EDC's existing processes and the CMDB schema, a CRUD matrix was created. See table 2 for an example.

In EDC's CRUD matrix, changes to data, including create, update, and delete, are initiated by working processes such as project management, procurement management and so on. Since the change management process will be used to control these changes, change management is excluded from the process column at this step.

Cis/Attributes	Processes	Product mgmt.	Project mgmt.	Contract mgmt.	Asset Mgmt.	Procurement mgmt.	HR mgmt.	Document mgmt.	AP dev. procedure	SLM	Incident mgmt.	Problem mgmt.
Software	purchase cost				C							
	purchase date				C							
	location		CU								R	R
	owner id				CU						R	R
	specification					CU					R	R
	asset id				CD					R	R	R
	SW name					C				R	R	R
	version					C				R	R	R
	type		CU							R	R	R
	vender						C				R	R
	purchase cost					C						
	purchase date					C						
	location		CU									R
installed on		CUD									R	R
owner id					CU						R	R
licence number						CU				R	R	R

Table 2 Example of a CRUD matrix

Developing integrated processes

Based on the output of a CRUD matrix, change management and configuration management are applied to the points at which CI data is accessed. In the EDC case, from the view of CI data, the process integrations were categorized into three types:

1. Automated data import - The corporate data is usually stored in application databases under the head office's control. There is a need for a pre-defined automation mechanism for collecting data from these databases. Contract management, asset management, and HR management are examples of this type. It is recommended that real-time or batch processing programs should be developed for data extract and data import because of efficiency considerations.
2. Data change control - Data that is changed by EDC's working processes needs to be controlled by change management. It is necessary to integrate these processes with change management, and slight modifications to each existing process are unavoidable. Examples of this type of process include product management, project management, procurement management, document management and application development management.
3. Data read access control - All EDC's processes, with the exception of the processes integrated with configuration management through the automation data collection mechanism, need to retrieve relevant information from the CMDB for specific purposes. The CI data is not allowed to be directly accessed from the CMDB. It relies on applications to present information to authorized users. SPMS, an application of EDC, provides user interfaces and a web portal for users and customers to get information about service operation.

These processes get CI information or make changes to CI through configuration management, irrespective of the types of integration. Configuration management is responsible for granting CI access rights to related processes.

A complete design output of this session can be found in the section "Control processes".

Control activities

The methods and techniques introduced in previous sections illustrate good practice in building a CMDB and developing integrated control processes. In this section, the activities related to the control of CIs are described in detail. In order to provide accurate and reliable information, CI data must be under strict control over its entire lifecycle. There are five essential control activities:

1. Define CI owners - CI owners are groups of persons, or specific processing programs, with the rights to make changes to designated CI attributes. CI owners are defined during the CMDB design phase, and their relationships with CIs should, preferably, be built into automation tools for enhanced control in the operational phase. According to the nature of updating CI behavior (See the section "Developing integrated processes"), CI owners are classified into program-based and task-performer-based owners, as follows:
 - Program-based owners - For CI attributes that are updated by automated data import programs, CI owners are assigned to the corresponding programs performing the changes.
 - Task-performer-based owners - If CI changes are initiated by RFCs under change management control, CI owners are assigned to the RFC performers. Normally a RFC is accomplished by a set of tasks. Each task is performed by certain types of professionals who are the CI owners.

2. Grant CI read access rights - Based on targeted users, CI data is divided into four levels, these being management level, administration level, operational level, and customer level. Each level represents a set of information that is readable for different user groups. The authorization of CI read-access is usually combined into the functions of applications, which are responsible for processing CI data and presenting information to different stakeholders.
3. Perform CI check-in/out - CI check-in/check-out control is an essential activity to maintain data integrity, by which one CI owner is allowed to check CI records out for modifications and other CI owners are not allowed to modify these records until they are in check-in state. Check-in/out activity is carried out with the concept of workspace, a temporary space, and should be undertaken together with change management.
4. Enable CI traceability - CI control should provide traceability of CI updates to prevent unauthorized changes. To establish the tracking mechanism that provides the basis of a configuration audit, it is suggested that the following aspects are covered:
 - Certain traceable information must be identified in advance. Although this information is normally part of the baseline data and the change records, it is the responsibility of configuration management to design and maintain an appropriate CMDB, including necessary tracking information.
 - CI update report should be generated regularly and should clearly describe each updated item and its associate information.
5. Prepare a checklist for configuration audit - Configuration audits are important activities to ensure the accuracy of a CMDB. Rather than performing periodic checks only for the physical existence of CIs, configuration audits have to be extended to include the process checks as well. During the design phase, EDC created an audit checklist, as shown in table 3, which contains part of the items to be audited.

Auditable items		Check (H/R)	Results
CIs	Attributes		
business services	name, version	R	
documents	name, version, location	HR	
servers	name, model, location	H	
software	name, version, location, license	H	
Check: Physical check (H), Process check (R)			

Table 3 Example of configuration audit checklist

Assignment of roles and responsibilities

The people factor is one of the critical success factors to ITIL implementation. Role specifications should be developed during the process design stage because designated staff have to participate in the process implementation. In this section, EDC's responsibility assignment table for configuration management is provided for reference. The column of "Assigned individuals" in table 4 refers to EDC's organization chart that was shown in figure 1.

CONTROL PROCESSES

Using the approaches introduced in previous sections, EDC developed a complete configuration control process integrated with existing working processes. In this section of the article, a simplified process is provided with a scenario description.

Role	Main responsibilities	Assigned individuals
The configuration manager	<ul style="list-style-type: none"> • The process owner • Define policies, CMDB scope • Identify CIs • Develop configuration management process • Monitor the configuration management process • Define management reports • Arrange training program 	<p>The leader of service provisioning department</p> <p>Selection criteria:</p> <ul style="list-style-type: none"> • Familiar with EDC services • Administrative skills
The configuration analyst	<ul style="list-style-type: none"> • Develop CMDB structure • Create CI control interfaces and procedures 	<p>The leader of platform development department</p> <p>Selection criteria:</p> <ul style="list-style-type: none"> • Ability to design relational database • System analysis ability
The audit team	<ul style="list-style-type: none"> • Prepare checklist for configuration audit 	The leader of platform development department
	<ul style="list-style-type: none"> • Carry out configuration audits 	<p>Staff in service provisioning department, and system department</p> <p>Selection criteria:</p> <ul style="list-style-type: none"> • Familiar with specific CI category
The CMDB administrator	<ul style="list-style-type: none"> • Maintain the CMDB and DSL • Manage physical CIs • Manage data access rights (account management) • Provide CI status reports 	Staff in service provisioning department
The CMS administrator	<ul style="list-style-type: none"> • The CMS owner • Design and maintain the CMS architecture • Evaluate tools 	<p>The leader of platform development department</p> <p>Selection criteria:</p> <ul style="list-style-type: none"> • Technical person • System analysis ability
The tools implementer	<ul style="list-style-type: none"> • Implement the CMDB and its management/ access interfaces by tools • Maintain the management platform of service processes • Provide technical support 	Staff in platform development department

Table 4 Example of responsibility assignment

A real example of a configuration control process

As described in section “Developing integrated processes”, EDC divided its processes into three groups, each with its specific integration method:

1. Integration through automated processing programs - The corporate data handled by contract management, HR management, or asset management will be collected by automated processing programs.

2. Integration through the change management process - Any changes to CI data initiated by working processes need to be carried out through change management. These workflows mostly remain unchanged, except for additional workflow of CI updates which are highly related to change management.
3. Integration through applications - When working processes need to retrieve CI data or generate reports from the CMDB, they will perform these activities through user interfaces provided by applications.

The simplified configuration control process of EDC is shown in figure 7. The ITIL processes that are not shown in the diagram were implemented by EDC after the configuration management implementation.

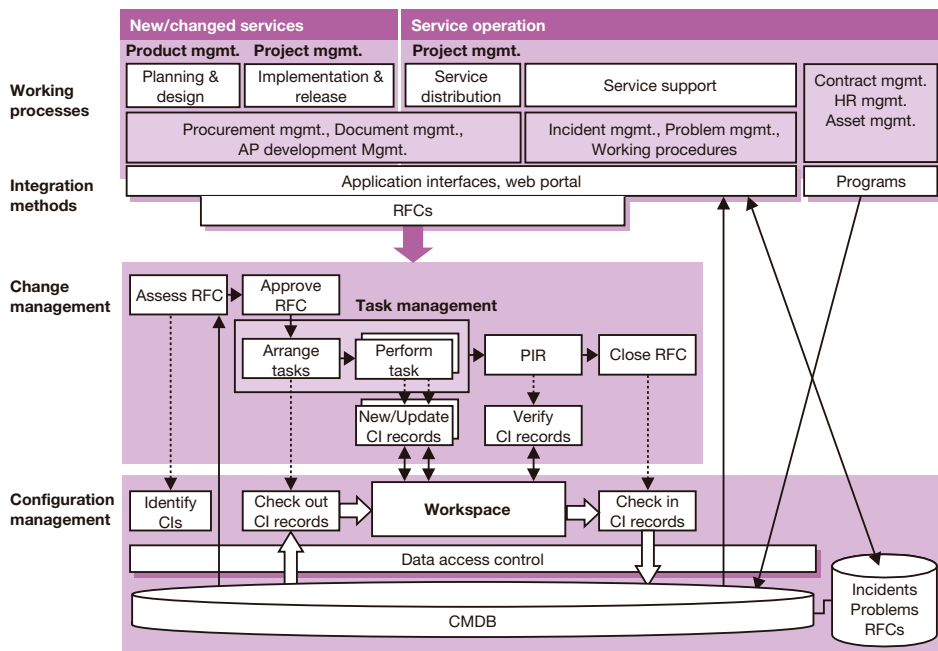


Figure 7 The simplified configuration control process in EDC

How it works with other processes

Two simple scenarios below provide an illustration of how the integrated processes work together.

Scenario 1: Developing a new service

EDC has assigned a product manager to develop a new service. He/she has to follow the following procedure:

1. Adhering to the product management process, the designated product manager designs the service content and creates a service catalog for this service.
2. Adhering to the project management process, together with associated processes, a designated project manager is responsible for implementing the service.
 - a. Initiate a Request for Change (RFC) to create a new business service in the CMDB. A change plan is provided at this stage.

- b. The CAB is responsible for assessing the RFC, where the Configuration manager is involved in the CAB to determine if the new service is within the scope of the CMDB.
- c. The approved RFC allows the project manager, as a task manager, to arrange tasks. Each task is associated with a set of CIs. In this scenario, tasks may include:
 - ♦ Build service basic information in the CMDB based on the service catalogue.
 - ♦ Purchase the required software and hardware, adhering to the procurement process.
 - ♦ Establish the operating environment.
 - ♦ Customize the service portal and reports, adhering to the application development management process.
 - ♦ Develop the necessary operational procedures.
- d. Tasks are assigned to specific task-performers according to a timetable. When tasks are assigned, the task manager has to check-out related CI records from the CMDB to a workspace. (Note: Task management is more complex in real life, since some tasks share common CIs or have to be executed in sequence.)
- e. The designated task-performer, also a CI owner, carries out his/her task and is responsible for creating or updating CI records in the workspace. All output documents are stored in the DMS according to the document management process.
- f. Once all related tasks have been accomplished, the RFC is completed.
- g. The change manager closes the RFC along with a check-in activity to check-in all related CI records after ensuring the RFC meets its objective.

Scenario 2: Moving a device from an alternate hosting site to a primary location

Incidents related to incomplete tape backup frequently occur in EDC's primary hosting site. The root cause of the problem is the overloaded backup jobs. In order to prevent the incidents from occurring further, the problem manager has approved a solution that involves moving a second tape library, originally located in the alternate hosting site, to the primary hosting location. To make the change to the IT infrastructure, the problem handler has to adhere to the following procedure:

1. Initiate a RFC for moving a tape library from the alternate hosting site to the primary hosting location. A change plan is provided at this stage.
2. The CAB assesses the RFC regarding its impact.
3. The approved RFC allows the RFC-initiator as a task manager to arrange tasks associated with specific CIs. In this scenario, tasks may include:
 - Remove the device from alternate hosting site.
 - Pack the device and deliver it to the primary hosting location.
 - Install the device into the primary hosting room.
 - Modify service document.
4. Tasks are assigned to specific task-performers according to the change plan. When tasks are assigned, the task manager has to check-out related CI records from the CMDB to a workspace.
5. The designated task-performer, also a CI owner, carries out his/her task and is responsible for updating CI records in the workspace. All output documents are stored in the DMS according to the document management process.
6. Once all related tasks have been accomplished, the RFC is finished.
7. The change manager closes the RFC along with a check-in activity to check-in all related CI records after ensuring the RFC meets its objective.

BENEFITS FOR OTHER PROCESSES

The CMDB as a basis for metrics of other ITIL processes

One objective of configuration management is to support all the other service management processes. Proper metrics are an important means of evaluating the effectiveness and efficiency of processes, and provide an input into decision support. Without a complete and accurate CMDB, process metrics will not be able to demonstrate their value. For example, if CI relationships are not well-established, an impact analysis report with incorrect information may be produced. There are many process metrics directly linked to a CMDB. Some examples include:

1. number of incidents/problems/RFCs by CI type
2. percentage of incidents handled within agreed resolution time specified in SLA
3. number of incidents caused by suppliers
4. report on CI usage against pre-defined thresholds
5. report on impacted area related to incidents/problems/RFCs/releases
6. report on operational cost by business service

Quality contributions

In addition to metrics generated from the information in a CMDB, there are some operational metrics that provide evidence of the contributions to quality made by an accurate CMDB. For example:

1. reduced incident resolution time arising out of appropriate levels of information held in the CMDB
2. accurate environmental information contributing to an increase in the percentage of successful changes/releases
3. better service level commitments made due to information on resources and capacity being more readily available for estimation purposes
4. increases in the percentage of changes completed on time because of proper allocation of resources
5. the availability of idled software license information when making new investment or software re-use decisions
6. cost/benefit analysis is more precise

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

In this article, several important issues were discussed. Business services are a starting point for the CMDB design. A service infrastructure analysis form, the Ishikawa Diagram, and guidance on defining CI attributes were all introduced in terms of assisting in the creation of the CMDB model. Another important subject covered in this article was how to design integrated control processes. Methods and techniques related to this topic include the CMS design and the CRUD matrix. Control activities are essential elements to the configuration control process.

Based on the methods and design concepts introduced, a configuration control process was developed and its simplified process was described in order to demonstrate the integration with existing processes. During the design phase, role specifications had to be developed. An example of responsibility assignment was used to describe role selections for the staff.

Recommendations

This article provided a systematic analysis and design approach applicable to the design phase of configuration management implementation. Its success in operation can only be guaranteed by following these further recommendations at the operation stage:

1. Use automated tools to make the process more efficient.
2. Carry out configuration audits and review audit reports for improvements.
3. Give awareness training and operational training to make sure all related staff are familiar with the processes and have a consistent understanding of these.
4. Review the CMDB scope constantly and ensure it remains appropriate.

Chiu-Ping Kuo (Taiwan) is an ITIL consultant for Acer Inc., where she is responsible for establishing ITIL processes and performing internal ISO/IEC 20000 audits. Chiu-Ping is also responsible for performing ITIL training course audits and reviewing ITIL exams as an auditor/reviewer for EXIN.

7.3 Bringing wisdom to ITSM with the Service Knowledge Management System

“Knowledge is a process of piling up facts; wisdom lies in their simplification” (Martin H. Fischer). The same can be said for the Service Knowledge Management System (SKMS): gathering data is one thing, but presenting the right information, at the right time, to the right people is another. Bryce Dunn and Linh C. Ho explain the what, why, how and who of knowledge management and the SKMS.

INTRODUCTION

It is often said that there is too much data and not enough information. That is meaningful information, upon which IT and the business can make informed decisions. Providing the right information to the right people at the right time empowers IT and the business to be knowledgeable and, in turn, be more agile in efficiently meeting client requirements while minimizing costs.

ITIL® V3 introduces knowledge management as a key process that touches all five lifecycle phases, but with particular emphasis within Service Transition. To effectively provide this information to appropriate IT or business users, a Service Knowledge Management System (SKMS) is required. This article offers an overview of:

- what knowledge management is
- why knowledge management and an SKMS are important
- how an SKMS can be achieved
- what the relevant components of the SKMS are
- what the differences and relationships are between the SKMS, the Configuration Management Database (CMDB), Configuration Management System (CMS) and asset repositories
- who should use the SKMS
- a case study using a real life example

When the quality of IT decision-making is improved, the value gained by the business from every dollar spent on IT is improved.

WHAT IS KNOWLEDGE MANAGEMENT AND A SERVICE KNOWLEDGE MANAGEMENT SYSTEM?

The goal of IT service management is to support business needs with technology using people, processes and technology. However, all of these elements need to be working together harmoniously in order to fully meet the business needs. The purpose of knowledge management and the SKMS is to help bring these components together to provide a central place for business-relevant information.

Without a central place for business-relevant information, IT groups with different technology functions operate in silos, where the focus is on the technology towers instead of

understanding the interdependencies and overall service affecting the business. Moreover, communication among the various IT groups, and with the business, is also minimal to non-existent. As such, when problems occur, time is mostly spent pointing fingers, which creates accountability issues and negative impact on the business.

The SKMS brings in all business relevant data from disparate sources that goes through the processing layer to turn these elements into more meaningful information for appropriate users. This helps IT groups to communicate between each other and, furthermore, enables various groups to gain an insight into the overall picture of how services are delivered and contribute to business success.

ITIL defines **knowledge management** as (Service Transition, page 236):

“The Process responsible for gathering, analysing, storing and sharing knowledge and information within an Organization. The primary purpose of Knowledge Management is to improve efficiency by reducing the need to rediscover knowledge.”

Activities of knowledge management include knowledge management strategy, knowledge transfer, data and information management, and the use of the SKMS.

The **SKMS** is defined as (Service Transition, page 244):

“A set of tools and databases that are used to manage knowledge and information. The SKMS includes the Configuration Management System (CMS), as well as other tools and databases. The SKMS stores, manages, updates, and presents all information that an IT Service Provider needs to manage the full lifecycle of IT services.”

WHY IS KNOWLEDGE MANAGEMENT SIGNIFICANT?

Decisions are made at all levels of the enterprise, from individual operators determining what course of action they should take to resolve an incident, to the CIO making decisions about investment priorities. Each of the different audiences needs up-to-date information available to them to assist them with making informed decisions.

The SKMS is significant because, if implemented correctly, it provides a single place for everyone in the organization to go to get the information that they need to make decisions that support the business.

One of the challenges that IT organizations face is the disconnect, not only between the business users and IT, but between IT management and the operators on the ground who are responsible for the day-to-day delivery and support of the service. The SKMS empowers these staff by providing them with relevant information about the individual decisions that they make and, in addition, with information about the wider context – for example knowing who is using the system right now, what their expectations are, what has been done before in similar situations and the results were.

Furthermore, IT organizations tend to operate in distinct “silos” based on technology rather than how they actually support the business. Each silo is responsible for one part of the technology “stack” and, in some cases, has little knowledge of the nature



Figure 1 An example of a SKMS presentation layer displaying relevant information for an insurance business service

of the overall service that is provided. The SKMS can help break down these “silos” by providing information that relates back to the overall system and facilitates communication. This contrasts with the commonly found situation where, if a critical issue happens, representatives from each silo gather around a meeting table for an acrimonious “blame game”. In other words, they can use the SKMS to understand the impact, the cause of the issue and help facilitate communication.

This problem is not due to a lack of data, rather it is exacerbated by too much data and not enough knowledge that is related back to the key decisions that need to be made. In this context, less is more; instead of huge amounts of raw data, it is desirable to provide information in the context of the decision that needs to be made. For example, a common scenario is that there are multiple issues happening simultaneously, so which one should the operator tackle first? To make the best decision here, the operator should ideally understand:

- The agreed business requirements:
 - When and where should the system be available?
 - What are any agreed performance targets or Service Level Agreements (SLAs)?
- The current business impact:
 - What users or customers are impacted by the service now?
 - Are there any revenue impacts or increased costs to the business?
- The potential business impact:
 - Are there any upcoming business events that would change the impact (e.g. start of trading hours for a trading system)?
- The nature of the system?
 - Do I have the skills needed to work on it?
 - Do I have the necessary access rights and permissions?

Instead of this level of information, in most cases operators are provided with volumes of technical event information that shows huge amounts of detail about different IT component failures, and cryptic symptoms logged in the service desk (“the network is down”), but nothing that will help them to understand the cause, the impact and the effect.

CMDB, CMS, ASSET MANAGEMENT AND SKMS – WHAT IS THE DIFFERENCE?

There is often a lot of confusion around the difference between asset management, configuration management, the CMDB and the SKMS.

Asset management is defined as (ITIL V3 Glossary):

“Asset Management is the Process responsible for tracking and reporting on the value and ownership of financial assets throughout their lifecycle. Asset Management is part of an overall Service Asset and Configuration Management Process “

So, in summary, asset management is all about tracking financial assets. While it is included in the asset and configuration management process (SACM), it is more of a financial management discipline. It does, however, provide useful information for decision making, so it can be viewed as information that should be available to the SKMS. For example, it may be useful to track hardware spending information for particular services to ensure that spending is prioritized for the most important services rather than the least. It would not be ideal for a

non- critical service to be running on fault tolerant hardware, while something that has the potential to have a significant business impact is running on non-redundant hardware.

A **CMDB** is defined to be (ITIL V3 Glossary):

“A database used to store configuration records throughout their lifecycle. The configuration management system maintains one or more CMDBs and each CMDB stores Attributes of CIs and relationships with Other CIs”

The CMDB is a data store for configuration management. Organizations today often have multiple CMDBs, frequently one for each area of their organization. The CMS brings the CMDB information into single view, using the federated model.

The main difference between configuration management and the SKMS is that configuration management is a static store of the configuration items (CIs) and the relationships between them. It does not contain live or historical data, and it is designed to describe attributes of the CIs, not provide a live system that shows not only the CIs and their relationships, but other knowledge that could be real time in nature. Configuration data is one of the major sources of data for the SKMS, so it can be said that the CMS is a subset of the SKMS.

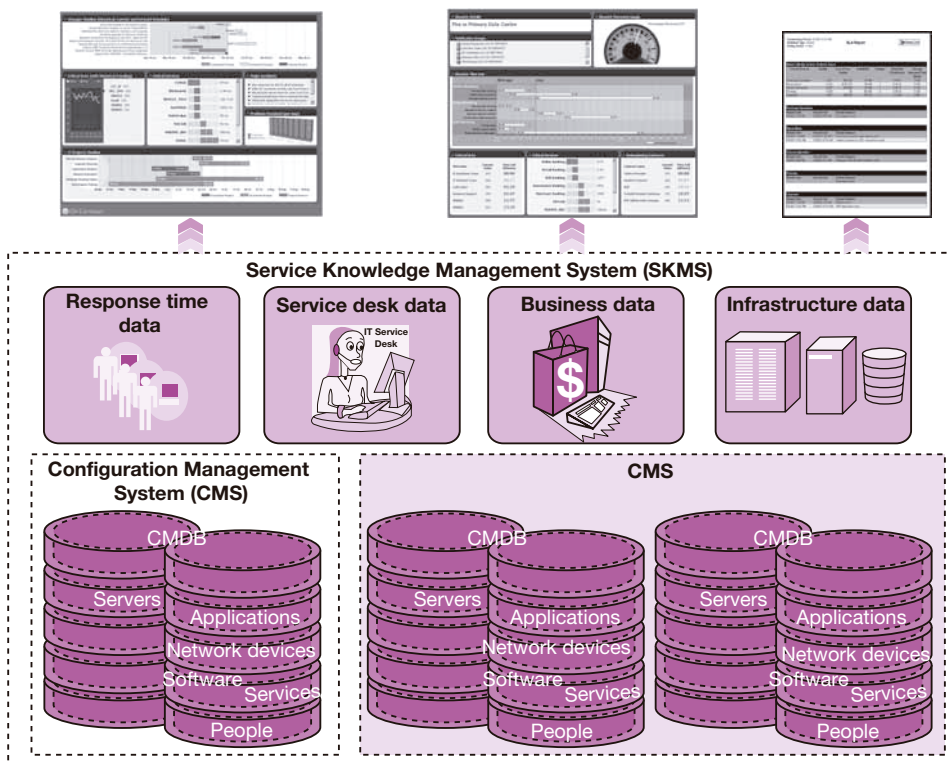


Figure 2 Example SKMS

Figure 2 shows an example SKMS, that takes information from multiple CMSs, which can each have one or more CMDB systems. In addition, it also takes data from other systems,

such as records from the service desk (e.g. previous problem resolutions, incident volumes, change management data etc.), business data (e.g. transaction volumes, revenue per system), infrastructure data (not only the configuration, but the performance), as well as response time data so that the experiences of business users can be ascertained.

So, in summary, asset management is focused on financial data, configuration management is a static representation of configuration data and the relationships between CIs, and the SKMS is a live system that brings data together from multiple sources for the purposes of decision making.

WHAT HAPPENED TO THE OVER-HYPED CMDB SOURCE OF TRUTH AND WHY DIDN'T IT LIVE UP TO ITS PROMISES?

One of the biggest areas of hype in the ITIL community over the last couple of years has been the CMDB. Vendors, industry analysts and others in the ITIL community were bullish about the benefits that could be realized by the implementation of the CMDB. This resulted in organizations embarking on “home grown” CMDB projects where one large database was created with the intention to populate it with all IT information. This strategy was doomed to failure from the outset. It quickly became apparent how hard it was to design and build a database that could contain ALL IT data, as well as keep it up-to-date and usable. Even the small number of these projects that were actually completed found that the data was generally out-of-date or not provided in a form in which it was easy to make use of it.

The industry has largely moved away from the one big database concept and has now embraced the federated model for the CMDB. With the federated model, instead of trying to design and build a large database to contain all information, the CMDB only contains some of the information, but has “links” to other data repositories to enable the rest of the information to be obtained on-demand. Essentially the federated model acts as a CMS that provides the ability to get information from many CMDBs as and when the user requests it.

However, while these systems do provide benefits by allowing many different IT systems (such as service desks, asset management systems, etc.) to relate their data, they have often not lived up to their promises. At their heart, the configuration management systems are still static repositories that focus primarily on infrastructure and asset information, and not on business information or business decision making.

While the CMDB was often touted as the single source of truth, the reality was often that most IT organizations ended up with multiple CMDB systems. Each different technology organization often went ahead and implemented their own different CMDB system, or one group (often the service support group) would try and forge ahead and create a CMDB system without executive buy-in, or buy-in from other organizations in the group. The result meant that each system had different sets of information (sometimes with some overlap) and it was difficult to determine where to go for the information. This only exacerbated the issue of keeping these systems up-to-date.

These systems are also typically static in nature, i.e. the system provides historical information about a particular CI, but not real time information about how that CI is supporting business needs, any current business impact and what can be done to mitigate any risk. A static CMDB on its own does not provide significant value, it is only when the CMDB comes to life through the SKMS that the true benefits are realized.

Ultimately, a CMDB is not the silver bullet to enable the successful implementation of service management nor is it a prerequisite. However, it can be a stepping stone to building the SKMS and/or CMS.

CASE STUDIES: KNOWLEDGE SYSTEM?

Case study 1: a financial institution

The following short case study focuses on a global IT outsourcer (ITO) responsible for a global bank's business. The ITO built an SKMS through a business service management (BSM) solution using a top-down approach. The key lessons to be learned from this study are truly understanding the needs of the business and customers, selecting key metrics relevant to the business, and undertaking ongoing measurement and reporting to prove the value of IT.

A number of challenges faced by the ITO resulted in them looking for an appropriate solution; these included:

- excessive and disparate data that was not of business-relevance
- lack of business understanding of their services and their relationships to the critical business processes
- domain expertise was in employees' heads, and was not documented
- problem resolution was based upon inefficient manual operational processes

During their initial strategy meetings, the ITO and the bank's executives met several times to understand the bank's critical business processes, their objectives and requirements. Moreover, IT also had to be prepared to set some realistic expectations and balance supply and demand of services.

Once the high level objectives of the bank were understood, captured and prioritized during the design phase, the bank and the ITO agreed that it would be prudent to start with one critical business service: check processing. Appropriate business users were interviewed to understand their daily pain points, objectives, and requirements. Key performance indicators were recorded for the check clearing process, and service level agreements were negotiated between the ITO and the bank. The ITO was the supplier of this service and was, therefore, responsible for the selection of the appropriate technology and people to assist with this project. The requirements not only extended to examining the check processing business process and determining the information needed, but also to providing this information in context to the identified business users. The outcome of this requirement for capturing and presenting this business information resulted in the need for the SKMS.

When the business requirements were gathered, the ITO was responsible for all transition activities including pre-production activities such as change management, configuration management, testing, and deployment. The initial project took six months to go live. All relevant data sources, ranging from configuration management systems holding federated CMDBs to service desks and infrastructure components, were consolidated to provide a unified view of the operation using a business service management solution. This combined both mainframe and distributed technologies. The critical business process was mapped to the underlying technology components using the service model in the BSM technology. The BSM technology provided the platform for the SKMS, enabling the collection of data elements, and providing the processing and presentation layers. Key business metrics and

business-oriented SLAs were configured to display on the dashboards as required for the tailored users. More importantly, each dashboard view was also secure. Each user group had different views for their respective day-to-day job. Users included both IT and business executives, directors, managers, administrators.

As the project scope evolved and expanded, 350 key business indicators were displayed across the SKMS presentation layer (dashboards) among seven different groups of users (both on the business and the IT side). This helped each user group make more informed decisions based on the information and wisdom provided through the knowledge system. The SKMS contained business information such as: number of checks processed, number of checks pending to be processed, and the monetary amount per check.

Results

The time saved by utilizing the SKMS enabled 50 full time employees to be reorganized so that they could work on more valuable activities to the business, such as improvement initiatives, rather than in their previous fighting fires roles. This global bank followed Six Sigma, and therefore a team was tasked to work with company Master Black Belts to improve other IT-dependant processes within the bank. 50% of time was saved by automating service level reporting, addressing accountability issues between the supplier (ITO) and the bank, and enabling quicker problem resolution time.

Case study 2: an IT service provider

Service providers are faced with the challenge of supporting multiple customers, each with different needs, different businesses and sometimes different services. This means that it is much harder to rely on inherent knowledge of the operations staff, particularly in a “leveraged” environment, where the same resources are used for multiple customers. When doing anything for a particular customer, service provider staff need accurate information about that customer’s particular needs. A good example is where the IT operations staff may be working on issues that affect customers. They need to understand exactly what service is being provided to the customer, what SLAs are in place and, ideally, information about the range of business processes they support for that customer.

These environments can also be extremely complicated as service providers often have different systems for each customer, sometimes contractually (where customers pay extra to ensure that their data is not on the same systems as other companies), or sometimes for historical reasons (e.g. taking over service desks from three different companies, with each implementing a different service desk system). This poses integration issues as well as the challenge of presenting the correct data.

This particular provider had an issue where they had multiple configuration management systems. The systems described different things and were not linked in any way. It was, therefore, difficult to get a complete picture of the configuration of the environment, and they were unable to provide real time information on top of the static configuration data to help them understand the service they were providing to their customers.

The first stage of their knowledge management project was to bring together information from the different CMSs for their different customers. This enabled them to understand much better the services they were offering and also the interdependencies between different customers, i.e. when there was an infrastructure problem, what the likely customer impact would be. This phase has been fully implemented and is in production.

The second step, still in the design phase, is to take this system and transform it into a true knowledge management system that integrates data from other systems including multiple service desks, infrastructure management systems, service level management data and, where available, business data. This allows them understand the complete service, not only in terms of what is offered, but also what is delivered on behalf of multiple customers. It also provides an excellent source of decision making data for this service provider's management.

WHAT IS NEXT?

Empowering business and IT users to have critical information at their finger tips (or at any given moment) helps transform a reactive and siloed IT organization into a collaborative and efficient service provider. This information allows each employee to understand the big picture, and every decision they make can be made in the context of the overall service and the business priorities instead of focusing on technical priorities. This level of focus on IT operations improves the service provided as it ensures that it is in line with business needs.

The SKMS can be an enabler to future improvement projects. Instead of focusing on improving IT from a technical or infrastructure perspective, the SKMS can be used as a decision making tool to select future improvement projects based on business needs. For example, looking at which business services were most impacted by IT over the last year and identifying the problems that are most commonly associated with these impacts. It is often hard to get good data about the true impact of problems, so being able to understand and quantify this means that decisions on improvement can be made that, in turn, provide the biggest ROI.

While the SKMS is valuable as a just a tool for IT, it really comes to life when business metrics are associated with IT metrics. Not only do the business metrics help show the real impact when there is an IT failure, but they also help to demonstrate the value that each IT systems brings. For example, it might be that an older legacy application is costing a large amount of money to keep running, yet brings in a small amount of revenue for the business. So the IT department could make a business case for it to be decommissioned.

Further improvements could be around predictive information. As the SKMS becomes the basis for decision making data, this data could be augmented with predictive and forecast type information. For example, based on previous history both short and long term, a service manager could be given a prediction that there is a 70% chance of an SLA breach in one month's time. Following on from this, it is likely that something could be done to save the SLA, thereby reducing the impact on the business and minimizing any potential penalty payment.

This same predictive information could also be used to analyze the data and make recommendations on the optimum times to carry out changes to the infrastructure that will have the least impact on the business based on certain criteria. For example the change manager may need to make a change in the next six weeks that will cause a one hour outage – based on SLA data and business impact data the SKMS could work out the best time for this change.

Bryce Dunn (USA), Senior Product Manager at Compuware, has been working in IT Service Management for over 10 years and has published articles and white papers on IT Service Management industry trends and best practices.

Linh C. Ho (USA) is Sr. Product Marketing Manager for Compuware's IT Service Management solution. Linh is co-author of the itSMF's Six Sigma for IT Management book and Pocket Guide, and was also on the review board for several itSMF books including ITIL V3 Foundations and Frameworks for IT Management. She is ITIL V3 Foundation certified and a Six Sigma Champion.

REFERENCES

Office of Government Commerce (2007). *ITIL: Service Transition*. London: The Stationary Office.

7.4 Starting right with the service portfolio & service catalog in the information age

***D**o you know what the single biggest error is that people make when they start out to make a service catalog? Have you ever seen one? Did you know that you should develop a service portfolio before a service catalog? In this article Mark O'Loughlin will provide advice and guidance on these topics and provide a useable template.*

INTRODUCTION - WE NEED A SERVICE CATALOG. MAKE IT HAPPEN!

It is inevitable that at some stage in an organization someone, hopefully management, will identify the need to know and understand what exactly it is that IT does, what it is that IT provides to the business, and how IT can help users help themselves. If they have an understanding of, or exposure to ITIL, then the requirement may be identified as needing a “service catalog” or in ITIL® V3 terms a “service portfolio incorporating a service catalog”. So far things are straight forward.

This requirement eventually lands on another manager’s desk who has just been given responsibility for the simple task of putting together the service catalog. Seems straightforward. If we just draw up a list of what IT does we are half way there.... Not quite. You are in effect half way to wasting a lot of people’s time, resources and efforts.

If you have never had to do this in the past, you certainly need some guidance to ensure that you:

- maximize the efforts of those involved
- get it right from the start
- design and implement a useable service portfolio and catalog framework
- design and implement something that will provide benefit to the business, including IT

In looking into how to go about this task there are a number of situations that can be faced:

- There is too much information on the subject and it is difficult to know what is what.
- There is too little information on the subject.
- One does not know where to start looking for accurate and concise information on the subject.

Suddenly the task seems not as straightforward as originally thought. Sound familiar?

In my experience, one of the most common mistakes that an organization can make is in mixing up the concept of what actually is a service portfolio and a service catalog. More often than not organizations start developing what they perceive to be a service catalog when, in fact, the exercise that they have just started is actually that of developing a service portfolio.

My intention in this global best practice article is to provide information on the basics of building a service portfolio framework which, in turn, will provide an input into the building

of a service catalog. I aim to distinguish between the two and explain how building a service portfolio correctly will only aid in the development and implementation of a service catalog.

Some organizations begin and never reach the end result, others begin and make a good start, and some do not know where to begin. There has been information published in the core ITIL V2 books but templates and step-by-step guidance has been lacking. Recent times have seen more information becoming available on building and designing service catalogs.

This “best practice” aims to guide the reader into a position to be able to design and develop a service portfolio that will support a service catalog. This guidance is intended for those who are about to begin their journey of designing and developing a service portfolio, or for those who have begun but would like assurances that they are on the right track. For those that are mature in their processes they may like to use this as a comparison for their implementation of a service portfolio. Either way, read on.

Due to the nature of what is expected and required of service catalogs, they generally require specialist application software to be fully deployed and utilized. Such software can be bespoke or off-the-shelf. Therefore I will focus more on providing guidance aimed at developing a structure for a service portfolio, which is pivotal when looking to setup and implement a service catalog. A service portfolio provides real value in the ability to be able to actually identify what it is that IT does and how it supports the business by providing IT systems that deliver business services.

In addition I will explore the relationship between IT, the business and, of course, the customers and users of IT. While there is a lot of focus put on the users and business defining the services that they consume, I want to make sure that this does not create the same disconnect that taking the traditional vice versa approach is said to do.

Configuring a full service catalog that encompasses all services that users need and that are offered by IT can be a mammoth task. My focus is on developing a service portfolio:

- which can be used in the absence of a service catalog
- aligns the Definitive Media Library (DML), Configuration Items (CIs), and Service Level Management (SLM) process
- records and creates relationships between IT systems and business services which feed into the service catalog

I will also link together the following:

- service catalog
- service portfolio
- Configuration Management Database (CMDB) and CIs
- DML/Definitive Software Library (DSL)
- Operational Level Agreements (OLAs) & Service Level Agreements (SLAs)

THE SERVICE PORTFOLIO

A common view of the service portfolio or service catalog is that, at its most basic level, it is just a list of all the services that are “consumed” by users within an organization.

Take this view at your peril as it is one of the bigger reasons for the service portfolio not providing any benefit back to the business. On its own a list of IT services does not allow:

- users to interact and order services
- business to understand how IT enables their operations
- chargeback for services consumed
- the ability for IT to know what it delivers in support of the business

It is important for management within IT to know what services IT are “enabling” i.e. the business services and what systems IT are providing i.e. the IT systems that support the business services. These key terms are described in full in the section “IT systems and business services”.

It is important for non-IT management to know what services are being provided to users, though this information can be at a high level (with a senior management audience in mind). This is so senior management can also have a view of what IT is enabling and supporting.

This can include services that are used but are not supported by IT. Examples of this are legacy systems. It is as important to capture what is NOT supported (and perhaps still being used) as well as what is being supported and used, so that support personnel know where to prioritize their time and efforts and management know what is in their remit.

If the service portfolio contains details of the services, where do you record the applications, software suites etc., which contribute to these services? The physical media that these are stored on would traditionally be recorded in a DSL. However we still need an inventory of the applications. These should be stored as CIs in the CMDB.

A service portfolio should be able to be built into any reasonable CMDB that can be customized. This has the further advantage of being made available to users of the CMDB. In addition, quick reports can be generated from the existing system detailing the services and related information.

THE SERVICE CATALOG

Once you have a service portfolio in place, you are in a good position to look at developing a service catalog. We have already established that by having a service portfolio in place you have a good understanding of what systems IT supports and provides, and what services are being used and required.

This information is essential to know. After all, the service catalog is at its very basic level an actual service provided to end users. This service should have the capability of allowing users to “order” services.

Basically a service catalog is some form of an interface that lets users select a particular service and have an interaction with that service. This could be requesting an email size quota increase. In this example the request may not be automatically approved or completed but, by the user interacting with the service catalog and requesting the quota increase, the request is triggered and sent to the appropriate people.

On-boarding of new personnel is another example of a service that can be provided by the service catalog. There can be many interactions that need to happen in the background between IT support teams and management (for approvals). However, all the end user should have to do is select what “on-boarding” services they want from the list of provided services,

“add them to the cart”, associate the interaction to the new user profile and submit the request. All the “magic” should happen in the background. The user should be able to see updates and progress on the request.

Typical traits of a service catalog are that:

- it is likened to the familiar shopping web sites with the ability to use a shopping cart
- it should be presented in everyday language that the user understands
- it needs to be easy to navigate the service catalog and for the user to find what it is that they are looking for
- it presents a list of the services that the users understand which are available for them to “order” e.g. a new laptop request, the ability for a user to release a blocked email themselves
- different users can have different levels of access dependant on their entitlement to a particular service
- it allows the user to see the status of their interaction and any history associated with it e.g. is it pending, approved or declined
- it provides management with a view of the services being used, who ordered the services and the number people using particular services

To recap:

The service catalog is an interface presented to end-users for them to “order” particular services that relate to them. The services are presented to the end-user in a manner that they can understand and relate to.

In order to build a service catalog it is **important** that you know:

- what services are being provided by IT
- what services enable users to perform business activities
- what are the services made up of in terms of CIs

This information is contained in the service portfolio. Knowing this information will help in understanding what are the services you can offer via a service catalog.

In order to provide a service to users to be able to update their email quota you have to be able to provide a reliable infrastructure to do this. That requires knowing all the CI elements. You need to know if you have the capacity (bandwidth) to provide this and if there is sufficient availability to ensure that the service can be delivered.

DEVELOPING THE SERVICE PORTFOLIO FRAMEWORK

A service portfolio framework should be:

- scalable
- easy to use
- low maintenance and low cost

A quality service portfolio:

- provides real benefit to both management and IT personnel
- is supported by an underlying process detailing roles, responsibilities and administration
- is linked to OLAs, SLAs and contracts, and contains business services linked to IT systems

Let us explore these concepts in more detail:

- **Scalable** - Can be used independently of any vendor software, however can also be used as the requirements specification and design template for creating a service portfolio using vendor software/applications.
- **Easy to use** - It should be easy to get information from a service portfolio. If support personnel or management cannot get the information that they need quickly, they will not use the service portfolio and there is no benefit to anyone in that.
- **Low maintenance** - The service portfolio should be maintained but the cost and overhead of this maintenance will be proportionate to the size of the portfolio and:
 - the number of services listed
 - the number of OLAs, SLAs and underpinning/service & maintenance contracts that are required
- **Provide real benefit** - It is of no use to any organization if the service portfolio does not provide benefit to the organization. The service portfolio should be able to provide basic information quickly and reference the underlying OLAs, SLAs and underpinning/service & maintenance contracts.
- **A service portfolio should not just consist of a list of services** - Do so at your peril, as all that will happen is that the list will remain a list since it:
 - will not be used by customers as they don't understand the "IT speak"
 - cannot be used to order services from
 - will not contain relationships to CIs within the CMDB and so it becomes an isolated, standalone, siloed IT list which, incidentally is the exact opposite of what we want in the first place
- **A quality service portfolio will be the direct link between the service catalog and the CMDB** - It will provide:
 - the portfolio of offered services
 - links to the supporting OLAs and any service and maintenance contracts
 - links to the supporting SLAs
 - an overall framework for understanding what it is IT does in support of the business, what it is that the business needs and consumes in the way of services
- **The service portfolio will provide a means to link:**
 - CIs to IT systems
 - IT systems to business services
 - Business services to service catalog services

In order to do this with the greatest of efficiency a service management application or toolset may be needed. However you still need to have design plans and templates created before implementing an application or toolset. Later on I will show you some examples of how to link all of these together with simple spreadsheet templates. These templates can be used as a service portfolio, or as the designs to be implemented into a service management system and/or service catalog application.

IT SYSTEMS AND BUSINESS SERVICES

Before developing a service portfolio we need to be clear on a number of key concepts and terminology. The ITIL core books provide details on these subjects. The following definitions are taken from the ITIL V3 Glossary which is now available.

Service catalog:

“(Service Design) A database or structured document with information about all live IT Services, including those available for deployment. The service catalog is the only part of the service portfolio published to customers, and is used to support the sale and delivery of IT Services. The service catalog includes information about deliverables, prices, contact points, ordering and request processes.”

Service portfolio:

“(Service Strategy) The complete set of services that are managed by a service provider. The service portfolio is used to manage the entire lifecycle of all services, and includes three categories: service pipeline (proposed or in development); service catalog (live or available for deployment); and retired services.”

So that's all we need to know? Well, not exactly.

Basically the service portfolio should provide information on the following:

- **Business service:**
 - is one or more IT capabilities that enable a business process or function
 - generally consists of one or more IT systems
 - are services used by the business in support of the business
 - something that a customer buys for users to use/consume
 - is seen from the perspective of the customer
 - is something that you can sell to a customer
 - is something that a customer will buy/is willing to pay for
 - is something that delivers value to customer(s) and users
- **IT system:**
 - is a grouping of CIs that make up an end-to-end IT solution e.g. a LAN
 - provides a capability to satisfy a need or objective e.g. the ability for all staff to access data
 - is built from CIs that exist in the CMDB via relationships between those CIs
- **Professional services/auxiliary services:**
 - services that can be provided by resources within IT (though not exclusively) that facilitate IT services & business systems e.g. business analysis, architecture & design, project management etc.
 - should be included under IT systems as a professional service if offered exclusively by IT personnel

Defining IT systems/professional services and business services goes a long way in knowing what IT provides users in support of their business objectives. Once you have identified the systems and services, you can quickly identify if you are in a position to support these or not.

Basically:

- Can you complete an OLA/SLA that covers the individual service requirements of the users and the business?
- Do you have support channels in place both internal and external?
- Is your service desk equipped to take and handle calls on these services?

How do you know what to charge for if you don't know the various components and CIs that make up the service. Ask any good accountant and they will tell you that you have to know every one of your costs before you can accurately cost a service for a customer.

LINKING SERVICES, CONTRACTS, OLAS AND SLAS

Now that we have identified the services and types of services, we need to look at OLAs and SLAs.

OLAs are agreements between IT departments detailing the activities and levels of support provided by each IT team in support of the service(s). OLAs will be related to the IT systems. If you achieve one OLA per IT system you will have identified:

- what you support
- what you don't support
- where gaps in support exist

To make things as clear and transparent as possible, I advocate one OLA per IT system. Metrics will not form part of the OLA but will be included in the SLA associated with the business service. An OLA can be as simple as detailing all the activities that are performed by the different IT teams in support of an IT system. It can include the hours of business for each support team and the expected availability of the IT system. Any differences between these hours need to be addressed as there will be a gap between actual support availability and expected support availability. This can be as simple as agreeing with the business that even though the expected availability exceeds the hours of support provided, the service will have no "out-of-hours support". As long as this is accepted and agreed with the business, on behalf of the customers, this is quite valid.

An SLA is a formal agreement between IT and the business to provide an agreed level of service. SLAs will be related to the business services. As with OLAs, work to produce one SLA per business system. If you have a lot of business services this can be aspirational. So as an interim measure introduce one overall SLA that covers basic operational and availability hours, and a single priority matrix etc. This gives you time to develop and agree individual SLAs as each service may very well have different support and availability hours, KPIs, response times etc.

The amount of OLAs and SLAs that you create may increase the administration needed to keep them accurate and up-to-date. It is vital to take this into consideration at the early stages of developing a service portfolio. Additional operational expenses may be required for maintenance of the service portfolio.

It will take time to develop all the required OLAs and SLAs so don't get disheartened if you are just starting out. After all, by this stage you should have identified the services. That's half the battle. Next step is to set up a program of work that will ensure these very important documents are completed, agreed, signed and added to the service portfolio.

PUTTING IT ALL TOGETHER

The following two pages contains the basic template for the overall service portfolio. This template can be used in a spreadsheet workbook, or can be the basis for designing CI record types in the CMDB. Either way individual worksheets or CI categories should be created for the following:

- DML
- CMDB
- IT systems & professional services
- business services

Due to the number of CIs an organization may want to track, it is unlikely that CIs can be maintained within a spreadsheet.

Figure 1 shows the DML and CMDB and the link between the two. The CI names from the CMDB will be related to the IT systems in figure 2. It is these CIs that actually make up the IT system(s).

Figure 2 shows the IT system(s) which is made up of a number of CIs. This relationship is important. You need to know what makes up your IT systems. This information is also valuable to change management in order to assist with impact analysis.

Figure 2 also shows the business service(s) which is made up of a number of IT systems, which in turn are made of CIs. In addition you will see that each IT system has an OLA, and each business service has an SLA. In the spreadsheet these are hyperlinked to the actual OLA and SLA documents so that they can be accessed easily and quickly from the service portfolio. No more looking around for that OLA or SLA again. An additional column can be added to store links to any service and maintenance contracts that are in place. These would generally be linked in the IT systems section.

The columns provided in these examples are the bare minimum that you should record in the service portfolio. I recommend adding in the following. This enables support personnel and management to find the details quickly without having to open the corresponding OLA/SLA/maintenance contract:

- level 1 support
- level 2 support
- level 3 support
- level 1 support hours
- level 2 support hours
- level 3 support hours
- agreed availability hours
- critical availability periods
- maintenance window for service

The use of hyperlinks in the spreadsheets allows the users to jump easily between the business and IT services. From within the business services there are the hyperlinks back to the IT systems. This is done by “naming” the IT systems service name in Excel and creating a hyperlink to it in the business services column called “IT systems”

From within each of the worksheets hyperlinks are also set up to the associated OLAs and SLAs, allowing a user to find the documents very quickly.

Infrastructure service	Communication service
Application service	Business service
Administration service	Database service
Security service	Professional service

Table 1 Types of services

DML

DML name	CI description	Location	DML type
MS Exchange SW	Email application software disk	DML library location 32B	Physical disk
Win 2000 Server SW	Windows server software disk	DML library location 03A	Physical disk

CMDB

	DML entries that are related to (installed on) the CI's		<i>Full support</i> <i>Decommissioned</i> <i>3rd Party support</i> <i>Requested support</i> <i>Under review</i> <i>Not supported</i>	
CI Name	DML relationships	CI description	Support status	Service type
Server 1	MS Exchange SW	Email server	Full support	Infrastructure service
	Win 2000 Server SW		Full support	
Server 2	Win 2000 Server SW	Domain controller	Full support	Infrastructure service
Switch 1	None	Core LAN switch	Full support	Support service
Router 1	None	Default gateway	Full support	Support service

See the next page where CI's are related to IT systems

Figure 1 The DML and CMDB Relationship

LINKING THE SERVICE LEVEL MANAGEMENT PROCESS

At a minimum we now have a service portfolio. If we have had the time and investment required, we could also have a service catalog (though this will generally come later). Regardless, the service portfolio needs to link into the service level management process. Governance is required for any new services that are being requested, and any updates and changes to existing services. Any changes to the service portfolio and service catalog should be subject to change control. This includes any changes to the status of services and any updates to OLA, SLAs etc.

A service portfolio can be developed at any time. Other ITIL frameworks do not have to be in place or embedded in order to start developing a service catalog. Change management, for example, will help maintain a stable environment in order to deliver the services but is not a prerequisite.

However, a service level management structure should be in place, even if it is in its infancy, before beginning to develop a service catalog and service portfolio. This provides the means to:

- maintain and develop the service portfolio
- assign ownership of the service catalog
- use service level management as a platform to build relationships between IT and the business (underpinning relationship management)

Once a new service is requested, the SLM process should have an action to make an initial entry into the service portfolio. The status should refer to the service being "requested". This ensures that we have an initial entry in the service portfolio so the lifecycle of the service can

now be tracked. Anyone looking for support on this service at this stage may only receive “best effort” as nothing has been defined or agreed at this stage.

Once done, the SLM process is responsible for identifying the Service Level Requirements (SLRs) that will eventually feed into the OLA and SLA for the service. These need to be negotiated and agreed. A formal OLA and SLA are put in place. Change Management is used to put the service into the production environment. After this, the SLM process should ensure that the service status is updated in the service portfolio and that all the related information is updated for that service.

I don't advocate putting in a service without having an OLA and SLA in place but it can happen. In such an instance you should detail exactly what it is that can be provided for that service and what cannot, and make sure that this information is added into the SLA and signed.

For example: if 100MB of bandwidth is expected but only 10MB can be provided due to a lack of investment in upgrading the current infrastructure, add this as a constraint to the SLA. Sign off that 10MB is all that can be provided at this stage. Failure to do so will mean that the users' and business expectation is for 100MB and IT will be perceived to be the cause of any bandwidth-related problems (or even just general problems).

MANAGEMENT OF THE SERVICE PORTFOLIO

By now we have a good framework for the service portfolio. A lot of work and effort has gone into getting to this point. However, like the CMDB, the service portfolio is only as up-to-date as the last update that was made to it. We don't have any mechanism or process by which to keep the service portfolio updated.

So we need a process to govern how to keep the service portfolio up-to-date. The process should include the following as a minimum:

- scope & objectives
- process flow
- process narrative
- roles & responsibilities

Scope & objectives

These are basic enough and additional details can be found in the core ITIL V3 books.

Process flow

There are two types of updates to the service catalog that may happen. Provisions need to be made for each. These are:

- updates that may be required from time to time
- updates that come from scheduled reviews

I favour splitting out these process flows into their own individual flows for the following reasons:

- It serves as a reference point.
- It reduces the complexity of the overall service level management process flow.
- It is easier for people to understand. They can just look at the specific flow for the situation.

Swim lane process flows can be developed from the ones presented here. Where possible, I tend to use this style of process flow for ease of use and identify who is responsible from a role perspective in the narrative.

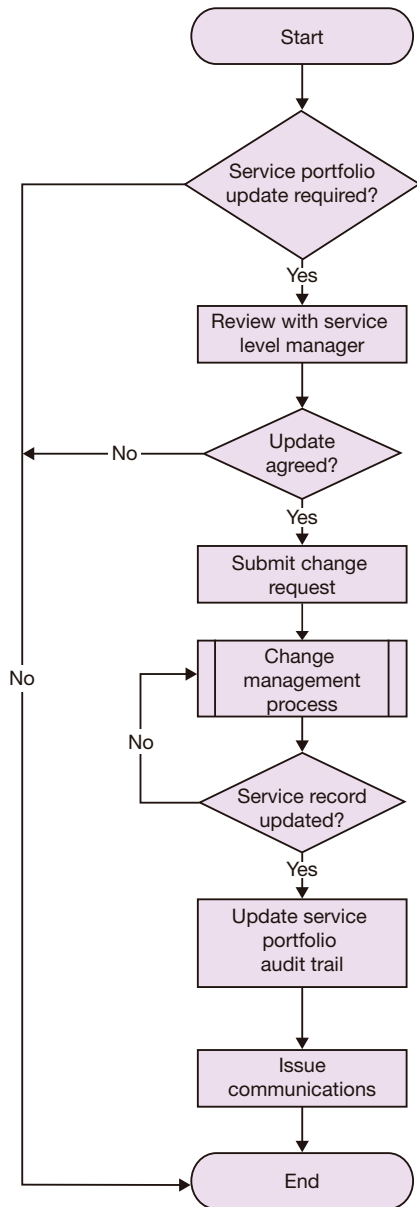


Figure 3 Service portfolio service update lifecycle

The main aim of this flow is to deal with service portfolio updates that are required before the next review of the service portfolio.

This is to ensure that before an update is made to the following, that the business and IT agree on the update

- Service portfolio
- OLAs
- SLAs
- Underpinning Contracts (UCs)

As you can see this is a very basic flow. But why make things complicated if they don't need to be?

This flow can be tailored to meet an organization's needs or used as is.

In this flow there is a review to be undertaken with the service level manager. This provides an input in the SLM process that should govern changing details in the OLAs and SLAs etc.

An example: if the business wants to change the support hours provided by the service desk, it is imperative that this request is reviewed by both IT and the business to ensure that:

- all parties understand the requirements
- IT can provide the new requirements
- the customers' needs are driving the request
- the business understands the impact to IT in providing the requirement
- the business is aware of the cost to IT of providing the requirement and, if required, are prepared to pay for the increase in service

The main goal here is to prevent assumptions being made and to ensure everybody is clear and has the same understanding.

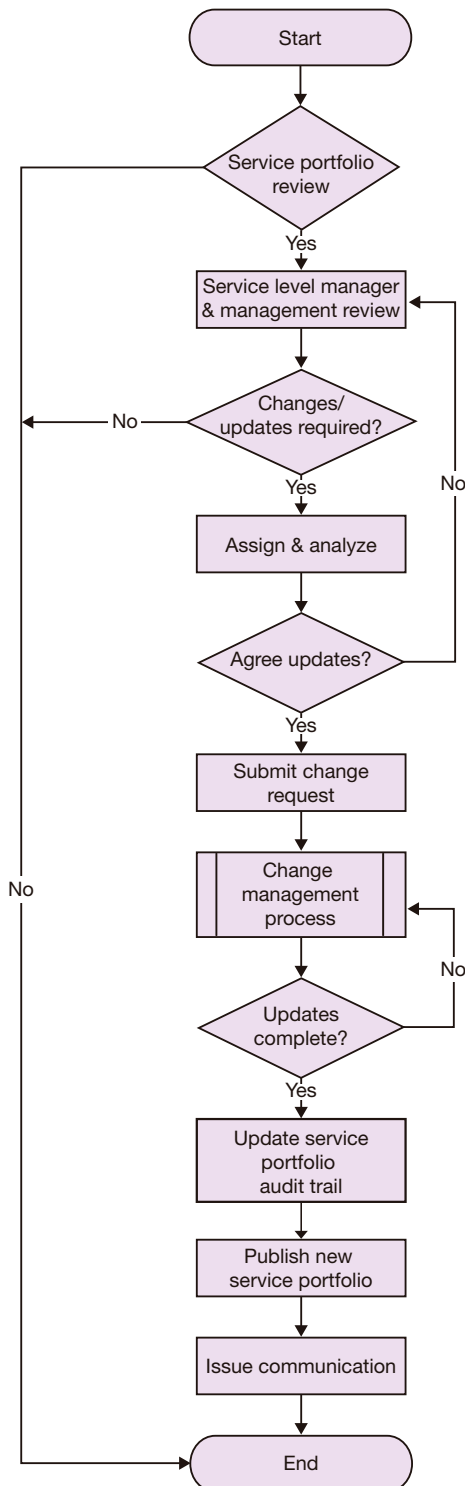


Figure 4 Service portfolio service revision lifecycle

The main aim of this second flow is to deal with a service portfolio review that should be scheduled on a regular basis, e.g. every six months.

This is to ensure that:

- the service portfolio is kept up-to-date
- the details contained in the service portfolio are accurate
- any changing needs of the customers and the business are identified

This flow can also be tailored to meet an organization's needs. It can be a reference for an existing audit schedule thus reducing additional auditing overhead.

Remember, the service portfolio, like anything else, is only as up-to-date as the last time it was updated!

The important points are:

- A review schedule is agreed from the start e.g. every six months, once a year.
- IT & business management (representing the users) are involved.
- IT & business management agree the updates.
- Any updates or changes are subject to the change control process.
- The updated service catalog is published.
- Any changes to the service catalog are communicated to the different layers within the organization.

Process narrative

The process narrative should list each process step and includes:

- the process step name and number
- the roles that can perform the process step
- description of the step
- additional details required to understand the step in detail

In the execution of the process, the support personnel involved will not want to have to read through a large process document to understand the process. However, for support personnel you can get them to focus their attention on just the process flow and the process narrative. This will greatly increase their awareness of the process. The rest of the process document is there to support the management of the process and provide the in-scope and out-of-scope boundaries for the process owner, process manager and anyone in the business that has a need to know.

Roles

It is vital to identify the roles within any process. Equally it is vital to ensure that these roles are actually assigned to people.

The roles should include:

- role name
- description of the role
- responsibilities required of each role
- basic level of skill and or/certification (optional)

It is also important to state that the roles can be performed by the same person or multiple people at the same time.

At a minimum, the service portfolio requires the following roles to support its maintenance and upkeep:

- service portfolio owner
- service level manager
- service portfolio librarian
- service owners
- IT support teams

Most of these roles will be required to support the SLM process.

List the roles and responsibilities in the SLM process. From the service portfolio management process, refer to the SLM process for further details of these roles. This reduces duplication between processes and makes it easier to update any role changes as you just update the SLM process only.

And finally, unveil the ugly duckling as a beautiful swan. It is important to publish the service portfolio. The service portfolio should be accessible, at a minimum, by all IT departments and business management.

The service catalog should be available to all users within an organization to allow them to “order” services. This may require a different interface than that of the service portfolio. Therefore the users will most likely see a different interface to that of the service portfolio.

CONCLUSIONS

In this best practice article I have set out to provide solid, real world experience in:

- understanding what a service catalog is and what it is not
- understanding what a service portfolio is
- providing a proven template for the basic design of a service portfolio
- describing the different type of services within an organization
- linking these services and their relevant OLAs, SLAs etc.

In my experience it is vital from the start to help organizations understand the concepts provided in this “best practice”. It is paramount to explore the differences between what is a service catalog, what is a service portfolio, and the role that the service level management process plays. Also, as is now becoming evident in the ITIL V3 material, the service catalog and service portfolio require additional processes to ensure that the information that they contain is:

- valid
- kept up-to-date
- still relevant
- provides benefit to users
- adapts to changes in users’ needs and requirements
- reviewed at scheduled times

Application technology is now available to provide a way of building a service catalog and presenting it to users in a way in which they can order services. However, as already discussed, it is vital to know what services are being provided to the organization beforehand i.e. IT systems and/or business services, which are the building blocks of, and make up in part, the service portfolio.

ITIL is a framework of best practices and can be adapted and adopted in many different ways. Organizations’ ITIL processes will differ in content and style but are designed to achieve the same outcome e.g. change management process.

As one organization is uniquely different to another, it is important to realize that the framework provided in this best practice can also be adapted to the organization and it’s capabilities. After all, that is what best practice is all about.

Looking ahead there will be an ever-increasing need for organizations to know and understand what services they need, and what services they provide in support of achieving and exceeding business goals and objectives. In a world where costs need to be reduced further, yet improved service levels are expected, the need for a service portfolio becomes quite evident.

In conclusion, if you are about to embark on service portfolio/service catalog project, or are progressing through one, I hope that the concepts and ideas expressed in this “best practice” offer guidance and clarity and a practical framework for designing and developing a service portfolio.

As a final thought, to quote the maxim “You cannot manage what you do not measure”. I prefer to look at this a different way “You cannot ‘affect’ what you do not manage”. Manage the service portfolio well to ensure that the services used by the organization are providing the best possible value at the best possible cost.

Mark O'Loughlin (IT Alliance, Ireland) is a consultant and specialist in ITSM frameworks including ITIL, MOF, COBIT, ICT Strategy and ISO 20000. Mark has designed ISO/IEC 20000 certified processes for multinational companies. Mark also writes about ITIL and ITSM and has been published in professional publications.

7.5 Service catalog management - Best practices and practical advice

As IT services are becoming commodities, IT must change the way it views its relationship with business units and end users. A critical step in this journey is the creation of a service catalog which defines and manages the relationship between IT and the business it serves. The authors, Bill Fine and Nick Schneider outline a step-by-step approach to designing and implementing an actionable IT service catalog.

INTRODUCTION

With the publication of ITIL® V3, the service catalog has taken on increased importance. In version 2, the service catalog was only briefly covered as an output of the Service Level Management (SLM) process. In version 3, the service catalog is promoted to its own process - and is recognized as the cornerstone of the service management lifecycle.

In practice the service catalog is becoming an essential element for a successful, service-centric approach to IT service management. In leading IT organizations the service catalog is used by IT to manage the service lifecycle, including service definition, configuration, continuous improvement and termination (what is referred to in this article as the “IT view” of the service catalog). The service catalog also is used to market available services to the business and to provide a means by which IT and its business customers can align around service demand (the “business customer view” of the service catalog). Finally, the service catalog is used as a means to manage day-to-day service requests, including online ordering and automated or orchestrated fulfilment (the “consumer view” of the service catalog).

So how do you create an effective service catalog? How do you get started in implementing the service catalog management process? What are some examples of IT organizations that have effectively produced a service catalog - and seen the benefits? This article will outline a step-by-step approach to designing and rolling out an actionable IT service catalog - while avoiding common pitfalls. The article will focus on real-world service catalog management as practiced by leading global IT organizations.

WHAT IS A SERVICE CATALOG?

Understanding the concept of a service catalog is best achieved by understanding the requirements behind its inception as well as the various purposes or views the completed service catalog supports. Both of these elements are examined further in this section.

Fundamental requirements for a service catalog

Organizations embarking upon a service catalog journey should be aware of fundamental organizational requirements that ultimately offer the richest form of success and business value, not to mention the most mature organizational capabilities.

A fundamental requirement is the awareness, desire and willingness to run IT like a business within a business. In this vein an organization is willing, able and ultimately required to:

- communicate service options with cost and performance choices, and deliver periodic performance reporting and reviews
- describe what they do in a manner which enables business to understand what they're getting, including costs and service level expectations
- compare their costs and service levels to the market, to determine what they do well and what they should consider outsourcing
- right size demand by giving the business choices through service offerings and service level options
- anticipate and respond more efficiently to changes in demand through the standardization of services
- bill or allocate all costs to business units, or the core budget, through the sale of products and services

More specifically, you need to start thinking like a general contractor and compete for the business of your end users. This will likely require making some fundamental changes to the way you design, market, price, and deliver IT services. This begins by establishing the customer-facing processes required to establish your portfolio of service offerings. These service offerings need to be customer-centric and business value-focused. See figure 1 for a description of the characteristics of a customer focused organization.

Traditional IT organizations	Modern IT organizations
Focused on technology	Focused on the customer
Firefighting mode	Demand driven
Organizational "stovepipes"	End-to-end process
Unknown costs	Financial transparency
Technical metrics	Business value

Figure 1 Managing IT as a service business

The easiest way to think about this mindset is to consider the responsibilities of an account executive at an outsourcer. This individual's role at the outsourcer is to understand what the customer wants, and to package together the component service offerings into a service portfolio that the customer sees value in and the outsourcer can provide at a profit. The portfolio needs to be priced in a way that makes sense to the business, so the business can manage spend, and it needs to be priced competitively.

The transformation to an internal service provider means that IT needs to be honest with itself about what services are offered as commodity services and best subcontracted to the lowest bidder, and what IT services are true value-add and provide competitive advantage.

The services that are best provided in-house stay that way and become your area of focus. For the commodity services, the fulfillment can be addressed by someone else. It is critical that you become the general contractor for the entire range of services, maintaining the relationship with your business unit executives and end users. This requires monitoring the service levels provided by your team or the outsourcer. Remember, if you give up ownership of the relationship with your customer, then you also give up the ability to demonstrate and drive business value.

In summary, fundamental requirements for service catalog success include running IT like a business within a business and thinking like a general contractor.

Multiple views into the service catalog

The best way to understand the concept of a service catalog is to understand the full range of views into it. There are two categories of views in the service catalog, with multiple types of views within each of these categories as described below. The respective roles and related processes will be discussed in more detail in the next section.

1. **Demand views** - Held by roles and entities that budget for, consume and/or order IT services. These break down as follows:
 - **IT view** - Held by product managers, service owners and process owners within the course of developing and publishing the service portfolio.
 - **Business customer view** - Held by business customers and relationship managers within the course of planning, budgeting and tracking technology spend.
 - **Consumer view** - Held by end users throughout the enterprise (including IT staff members) within the day-to-day course of requesting and consuming individual services.
2. **Fulfillment views** - Held by roles or entities that deliver or fulfil IT services:
 - **Functional view** - Held by service owners within the course of delivering contracted functionality.
 - **Support view** - Held by process owners within the course of delivering contracted support.

See also figure 2. We'll consider these views in greater detail in a later section.

THE SERVICE CATALOG MANAGEMENT PROCESS

Underlying every successful service catalog are a number of key roles, processes and automated capabilities or design requirements. This section explores these in greater detail.

Service design requirements

Based on the views and respective roles included above, there are a number of service design requirements that are core to an effective service catalog. These requirements are broken out from both demand and fulfillment perspectives:

- **Demand requirements** - The service catalog must be able to:
 - set appropriate expectations through effective service names and descriptions
 - communicate and manage service level commitments through pricing and Service Level Agreement (SLA) metrics
 - support efficient searching through categories, keywords and icons
 - streamline the ordering process through auto-fill forms
 - facilitate order completion through service specific forms

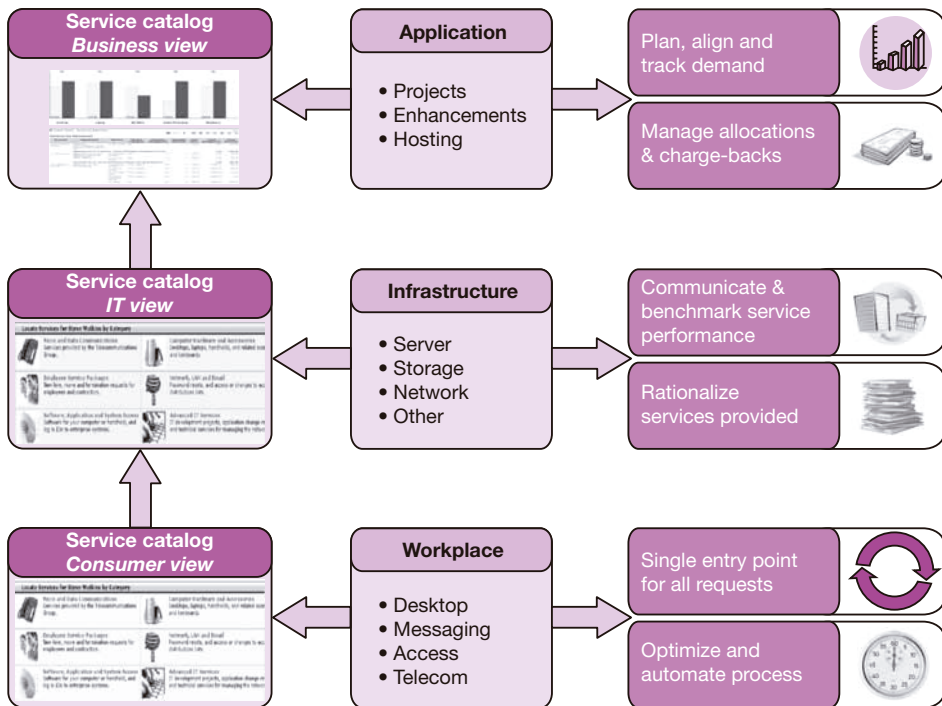


Figure 2 Service catalog demand views

- enable correct and accurate orders through appropriate field level instructions
- offer real time status updates through mechanisms such as status email templates
- **Fulfilment requirements** - The service catalog must be able to:
 - recognize organizational design through service teams and work queues
 - support governance through standard authorizations
 - drive consistency through standard delivery plans
 - ensure quality through accurate checklists
 - enable efficiency and speed through intelligent workflows
 - integrate automation through task level APIs
 - set appropriate expectations through service level standards

Service catalog roles

A number of roles were introduced in the previous sections. These roles, among others, are now discussed in more detail below.

- **Relationship manager** (also commonly known as service level manager and account manager) - The relationship manager is responsible for the relationship between IT and the business. It is the relationship manager who is responsible for working with the business owners to understand requirements, and to establish and document service agreements based on the catalog of services provided by IT. Additionally, this role is responsible for the creation of customer-facing service reports which provide the business customer with visibility on how well their services are being delivered according to the commitments and agreements made in the service catalog and related service level agreements. Based on IT's performance, the relationship manager gathers feedback on business customer satisfaction and anticipates customer demand. Further, this role handles customer

complaints and shields the business customer from IT complexity, and ultimately identifies opportunities to create new business value from IT.

- **Product manager** (also commonly known as service delivery manager or portfolio designer, and sometimes a shared account manager role) - The product manager is accountable for the overall service offerings published within the service catalog as well as the review and continual improvement of those services. The product manager is responsible for assembling services that cut across different functions and silos and provide offerings that make sense to the business customers. This role is also responsible for establishing pricing, defining service levels, analyzing competitiveness relative to the marketplace and providing recommendations on how the service should be sourced. Product managers work very closely with the service owners who provide specific services within the service offerings that the product manager has responsibility for.
- **Service owner** (also commonly known as service manager) - Service owners are responsible for specific services regardless of where the people and technology resources that build and maintain the services reside within the environment. This role is ultimately responsible for managing changes to the services for which they are responsible, as well as the continual improvement of these services in line with customer requirements, whether these are requirements from the product manager, business customer or otherwise.
- **Process owner** (also commonly known as process manager, support manager or service manager) - Equally important to service ownership is establishing process owners for the processes that cut across multiple functions or silos. Similarly, process owners are responsible for managing changes to their processes as well as continual improvement. Given the horizontal or dotted line nature of their role, process owners (and service owners) can only be as effective as the levels of priority and empowerment that have been assigned to these roles by IT executive management.
- **Communication specialist** - Given the organizational transformation required to enable maximum service catalog success, communication specialists are required to help plan and execute communication and awareness programs. These programs must be incorporated at an early stage into the overall service management organization change management strategy. Typically, this skill set resides outside of the IT organization, and communication specialists can typically be found in corporate communications and marketing.
- **IT financial analyst** - IT financial analysts are responsible for supporting product managers in establishing pricing models, setting pricing and subsequent P&L analysis (if applicable). IT financial analysts additionally serve as a critical interface during the ongoing accounting and financial forecasting activities. As with communication specialists, this skill set sometimes resides outside of the IT organization but there is a continued trend toward establishing an IT finance executive role reporting to the CIO.

These roles are associated to the related processes in the following section.

Related activities and processes

Underlying these roles are common processes and activities that go hand-in-hand with successful service catalogs. These processes are summarized below with an ARCI (Accountable, Responsible, Consulted, Informed) role analysis.

- **Service portfolio management** includes the following activities: defining, standardizing, publishing, communicating and updating the service portfolio. Product managers are primarily accountable for these activities and share responsibility with service owners, while relationship managers are consulted.

- **Financial management** includes the following activities: budgeting, accounting and charging. Product managers and relationship managers are accountable for these activities, while the IT financial analyst is responsible and corporate finance is consulted.
- **Demand management** includes the following activities: influencing demand, forecasting demand, measuring and communicating consumption, and accounting/adjusting for variances between forecasted and actual consumption. The relationship manager is accountable and shares responsibility with product managers, service owners and process owners, while the business customers and end users are consulted.
- **Service level management** includes the following activities: negotiation (includes creation, agreement and application of SLAs), reporting and communication, and continuous improvement (including consequences). The relationship manager is accountable and once again shares responsibility with product managers, service owners and process owners, while the business customers and end users are once again consulted.
- **Service catalog management** includes the following activities: defining, standardizing, publishing, communicating, and updating the service catalog. Service owners are primarily accountable for these activities while subject matter experts (SMEs) are responsible and product managers are consulted.
- **Supplier management:** depending on an organization's sourcing strategy, these activities sometimes fall within the service portfolio management and service level management processes, with accountability and responsibility shared by product managers, relationship managers and vendor management, whilst legal is consulted.

Importantly, in all cases IT executive management is kept informed. These process activities are explored in more detail in the following section.

A STEP-BY-STEP APPROACH TO A SUCCESSFUL IT SERVICE CATALOG

Having established the various service catalog views, roles and processes, the combination of the three are now brought to bear below in a step-by-step approach to setting up a successful service catalog.

The service portfolio - evaluating potential service offerings

Relating this step back to our views, the focus here is on the IT view of the service catalog. By working with service owners and process owners, this view is driven by product managers for the benefit of relationship managers.

Firstly, it is necessary to review the current state environment to understand the current services being offered, and the supporting environment to include current strategies, processes, roles, technologies and contracts. Current state maturity and desired maturity are determined and factored into future resource requirements planning. Out of this review, strengths, weaknesses and improvement opportunities are identified and factored into an implementation strategy and roadmap which prioritizes customers, their respective services and establishes roll out timing. In order to most effectively roll out their portfolio of service offerings, product managers must then establish their service portfolio taxonomy which will govern the design decisions and the ultimate construction of service offerings. The taxonomy consists of:

- formal model decisions which include:
 - shared vs. dedicated components (i.e., infrastructure)
 - service segmentation

- marketing approach
- management guidance
- standard packaging conventions which include:
 - service offering specifications
 - naming conventions
- enterprise list of service offerings which includes:
 - names
 - short descriptions
 - categories

With the taxonomy established, product managers review current costs and establish their pricing approach as well as their service level approach. The service offerings are then built, incorporating:

- service level objectives
- price
- component services
- service categories
- management details

Through the benefit of this IT consumer view of the service catalog, product managers are now ready to publish their portfolio of service offerings for consideration by the business customers and their relationship managers.

Demand management - managing IT demand through the service catalog

As we move from the service portfolio step to the demand management step, we also shift our focus to the business planning view of the service catalog. Leveraging the published portfolio, this view is driven by relationship managers for the benefit of their business customers. More specifically, by acting on their unique understanding of their business customer's requirements, as well as the specifics of the service portfolio, relationship managers are able to present appropriate choices to their customers with regards to cost and service levels as they budget for the year ahead. These choices, in effect, serve as knobs which the business customers can tune as necessary to ensure that their technology spend remains in line with their ever-changing business needs and demands. As a result, relationship managers are well positioned to manage and make adjustments for changing demands at the business planning and budgeting levels.

Further, the component services within the resulting service agreements are related to the orderable services that will be consumed by staff members within the business customer's business unit(s). This empowers relationship managers to manage demand at the consumption level through a variety of means, which range from entitlement (governs who is entitled or authorized to order services based on the agreement) through to tracking and periodically reporting on the variance between planned and actual consumption.

The negotiations and subsequent agreements enabled by this business planning view of the service catalog trickle down to the orderable request level which relates to our remaining views. From the demand perspective, this is the consumer view of the service catalog, and from the fulfilment perspective this relates to the functional and support views of the service catalog. These are explored in more detail in the next section.

Request fulfilment - requesting services through the service catalog

In this step, the consumer view of the service catalog is driven by service owners for the benefit of end users. The service owners design, configure, test and deploy orderable services for each of the component services included within the product managers' service offerings. Based on their business unit's respective agreements, end users are entitled to request and consume selected services for their business purposes.

Behind the scenes, the service owners hold a functional view of the service catalog thus ensuring all necessary functionality and component services are delivered as negotiated in the service agreements. Likewise, the process owners hold a support view of the service catalog, thus ensuring all delivery and support service level objectives within the service agreement are understood and delivered upon.

Continual improvement - increasing efficiency and service quality

Continual improvement is our final step. This step is driven by product managers, relationship managers, service owners and process owners for the benefit of business customers and end users. This is an ongoing activity based on KPIs and the service level metrics established in the processes highlighted above. Ultimately, metrics are aligned with the service level objectives within the service agreements. The metrics focus on a number of areas:

- service quality and customer satisfaction
- timeliness and cost efficiency of the internal delivery
- performance of third party suppliers
- compliance with contract commitments and regulatory requirements

The cumulative output of these metrics helps product managers, relationship managers, service owners and process owners to identify opportunities to improve services, based on enhanced IT capabilities and an ever-increasing understanding of their business customers and the associated business drivers.

Going to back to the notion of running IT like a business within a business, and thinking like a general contractor, the criticality of continuous improvement is easily understood and applied to future planning considerations.

In the next section, we'll examine service catalog examples from IT innovators and industry leaders.

SERVICE CATALOG EXAMPLES

As indicated in the prior sections, an integrated service catalog supports a variety of different views or purposes. What each of these views or purposes has in common is, firstly, that they share and take advantage of a common set of service definitions maintained in a single service catalog. Secondly, they make clear that the service catalog is not an end itself, but rather exists to support and enable a variety of processes and activities required of a customer and service focused IT organization. In this section we describe in more detail how a service catalog can be put to use in support of three related processes: service portfolio management, service demand management and service request management.

Example one: Service portfolio management

Most large organizations manage and support hundreds or even thousands of business applications and their related infrastructure. The cost of this support typically represents

70% or more of a typical IT budget. One Fortune 500 company we are familiar with used the service catalog to define and track the components, cost and service level performance of its varied business applications. It then used this data to work with the business to assess which of the legacy applications continued to drive meaningful business value and which could be terminated. The results of this effort were that 5% of the portfolio of business applications were terminated with the consent of the business. As they go forward, each business application is continuously assessed against business value metrics, in just the same way as any newly proposed project or initiative.

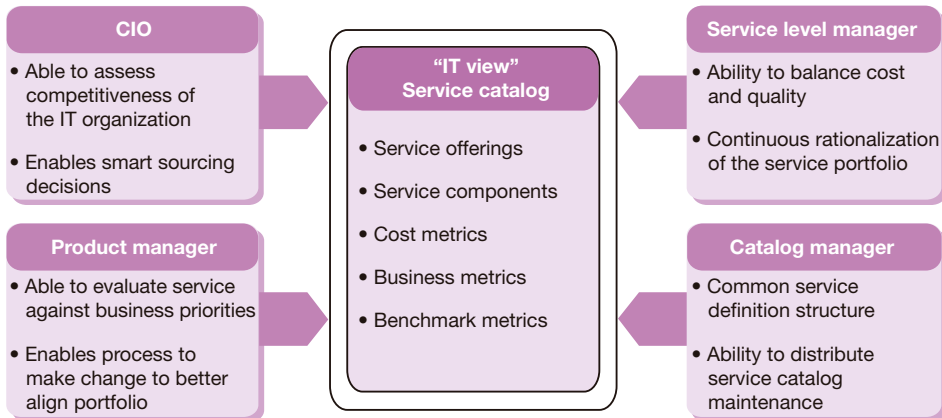


Figure 3 Service portfolio management

Figure 3 describes the key attributes of the "IT view" of a service catalog, and the roles it serves, together with a description of how each role utilizes the service catalog. The key theme that runs consistently through each of these roles and their use of a service catalog is one of service portfolio optimization. By defining what it does from a services perspective, an IT organization provides itself with a foundation for assessing what it does well, what it should continue to do, what it should outsource, and what it should quit doing entirely (because it either no longer drives business value or is redundant). By defining the services from a business perspective, the IT organization gives itself the means to engage the business in a dialogue that will enable better alignment as these choices are made.

Example two: Service demand management

Extending the service catalog to support a business unit's specific planning and budgeting processes is the next level of maturity. Figure 4 describes the key attributes of the "Business view" of a service catalog, and the roles it serves together with a description of how each role utilizes the service catalog. In this business orientated view of the service catalog, what a business manager or relationship manager typically wants to know are answers to the following questions: What is my group consuming? Is this more or less than what was planned? Am I getting what I am paying for? Am I paying for some other group's consumption? What can I do to lower my costs? What can I do to get better service?

Another Fortune 500 company we are familiar with determined that they needed to find a better way to allocate application infrastructure and support costs across business units that consumed shared IT resources. This particular organization decided to initially leverage

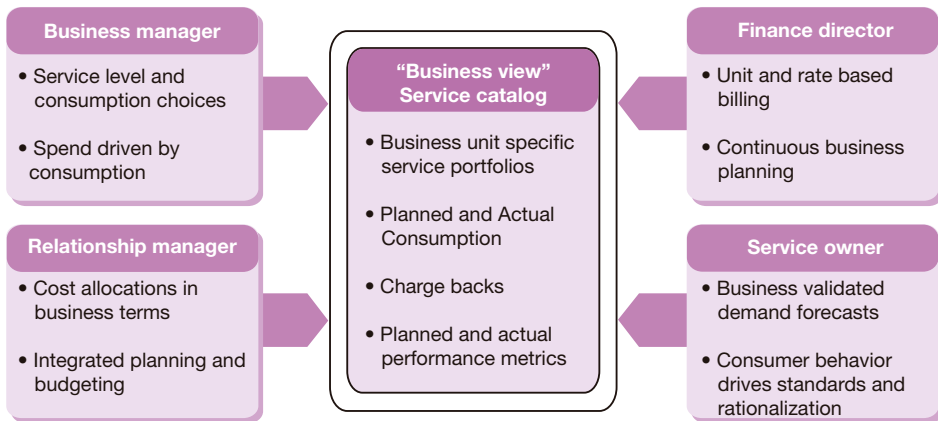


Figure 4 Service demand management

their service catalog in order to drive business unit-specific agreements regarding the planned consumption of application maintenance and enhancement resources. Instead of determining in advance the “proper” amount of IT resources allocated to maintain and enhance a shared application, and then distributing these costs as a flat tax across all business units using a particular application, this organization used the catalog, linked to the budgeting and planning process, to ask the business units to subscribe to and pay for such resources on a consumption basis.

As an example of the kind of changes this approach enables, this particular organization’s typical application maintenance and enhancement budget for SAP was roughly ten million dollars. By moving to such a demand-based approach to planning, certain business units will end up paying more than they had in the past, and be happier because they will get more enhancement support than they had in the past for services that they determine are important to the operation of their business. Other business units will get less service, and be happier because they will pay less for what they perceive to be non-critical services. The end result is that all are expected to be happier and the overall IT budget for this expense item is expected to remain the same or decrease. Here this organization is using the catalog to enable the business to drive spending. The next step beyond application maintenance and enhancement spending is to apply the same approach to other variable service components, such as storage service levels.

Example three: Service request management

Either governed by business specific agreements, as described in the section above, or independently, the third example of how to effectively leverage a service catalog to drive business value involves using the service catalog to drive the request fulfillment process. Figure 5 describes the key attributes of the “Consumer view” of a service catalog, and the roles it serves together with a description of what each role does with the service catalog.

There are two key concepts that drive the consumer view of the service catalog. The first is that consumers of IT services (e.g. employees of all sorts, project managers, etc.) expect that their IT organization should interact with them and their requirements in the same manner that they have become familiar with in their non-work lives, with companies such as Amazon, eBay, etc. The second is that once orderable IT services are catalogued, their fulfillment

processes can be defined and automated, driving efficiencies and predictability in the service delivery process.

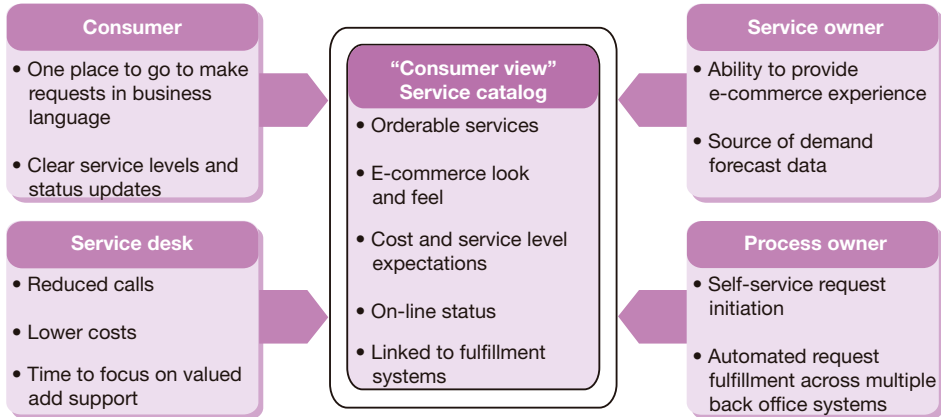


Figure 5 Service request management

The final Fortune 500 Company we will profile here was faced with a situation where IT service delivery could not keep up with the growing business demand. 41% of services were delivered late and employee waiting time at corporate headquarters was estimated at three million hours per year. The answer was to adopt a consumer orientated service catalog that provided:

- an intuitive web-based interface for employee self-service
- a single, integrated platform across over 500 IT, telecom, workplace, and facilities services
- best practice work plans to reduce service fulfillment cycle time
- automation of service delivery processes, easily configured services
- reports to measure performance, achieve continuous improvement

The resulting solution provided a “new face” for IT and delivered the following business benefits:

- reduced waiting time by 1 million hours = increased internal customer satisfaction
- shortened cycle time by 30%, with 90% of requests delivered on schedule
- reduced number of IT process “shepherds” (service coordinators) by 75%
- cut service delivery costs by 37%, with 3 year ROI of 300%

FOUR COMMON SERVICE CATALOG MISTAKES - AND HOW TO AVOID THEM

Inspired by success stories such as those chronicled above, and driven by the desire to incorporate ITIL best practices, organizations are rushing to create and deploy service catalogs. Many will succeed, while others will undoubtedly fail to achieve the full potential of a service catalog. Drawing on our experience of working with hundreds of companies at various stage of a service catalog project, this section outlines some of the most common mistakes made in the development and deployment of a service catalog initiative.

Mistake one: defining services from the IT perspective

A common mistake with many service catalog initiatives is defining services in technology terms, with service levels based on the metrics that are easiest for IT to track. We call this the inside-out approach and it almost always fails. Successful service catalog projects start by asking users and business stakeholders what they want and what's important to them, and then building the catalog around those success factors. This is the outside-in approach.

The problem is that while IT tends to be organized around technical, skill-based or asset-based silos, business users think in business outcomes. So while IT's customers may be thinking about on-boarding new employees or their order-to-cash process, IT is talking about their change management process or distributed computing.

If you package and communicate your services and metrics with a focus on business-relevant deliverables, rather than the underlying technologies and technical service levels, you've overcome one of the greatest barriers to success.

Mistake two: assuming that a list of services is a service catalog

Recently, the IT infrastructure group of a large corporation spent two years implementing ITIL. They sent their entire team on ITIL training; they dutifully documented their processes and catalogued their services. But nothing changed in their interactions with business users. Despite having a list of IT services readily accessible on the corporate intranet, business users didn't seem to want to refer to it.

The problem was that their service catalog was merely a static reference document. End users could go to it to read about IT services, but they needed to link to another form or call the help desk to submit a request. Business unit executives could skim the document to see service level commitments and budget allocations, but they had to contact a relationship manager for up-to-date information on service performance, cost, and quality. The service catalog became just an extraneous step in the process.

Too many service catalog projects stop with simply publishing a document or posting it on the Web - these catalogs aren't used and they don't make an impact. A successful IT service catalog is accessible at the **moment your customers want to** or need to think about IT.

Here are a few tips:

- **Make it interactive** - Rather than presenting a super-set of all possible services and options, personalize the users' view of the service catalog based on their job function, location, and role.
- **Make it actionable** - End users should be able to place an order within the service catalog, and business executives should be able reference the service catalog when they want to review their IT bill or make sourcing and budgeting decisions.
- **Keep them coming back** - Use the catalog to keep the business updated on IT consumption and service levels, and provide end users with the ability to check the online status of their requests.

Keep these principles in mind and you can overcome this pitfall and promote adoption. Otherwise, your service catalog is likely to gather dust on the virtual shelf.

Mistake three: thinking of the service catalog as an extension of the service desk

Many failed service catalog projects start at the service desk. At first glance the service desk may seem like a good place to start. Following ITIL guidelines, the service desk is the point

of contact with your end users for addressing any problems, complaints, or questions. But based on experience, it is clearly not the right place to start with your service catalog.

First of all, the service desk is designed around incident management to address service disruptions - whereas a service catalog should be focused on ongoing service operations and service agreements to support the business. Moreover, the service catalog encompasses more than just services associated with the traditional help desk. The purpose and scope are different, and the approach should be different as well.

Another challenge associated with the service desk approach is the interaction model. In an effort to make the service catalog actionable and transactional for end users, I often see service descriptions linked to a self-service help desk form. Unfortunately, the design of these forms is typically based on a CTI (category-type-item) and trouble ticket structure. The CTI structure works well for trained staff to quickly record and resolve incidents. But for business users, it is extremely cumbersome to choose from a pick list in a CTI structure; in many cases, they are faced with a series of cascading menus that fill the screen. End users simply refuse to adopt this self-service model; instead, they pick up the phone to call the service desk. If you had hoped to use the service catalog to drive standards and reduce costs by eliminating manual steps - yet you approach it as simply a front-end to your existing service desk system - you clearly won't accomplish your objectives.

To succeed in making your service catalog actionable and user-friendly, you need to provide business users with an interface that they are familiar and comfortable with. Look to the leaders in e-commerce for ideas. Provide the same look and feel (e.g., search and browse, create a shopping cart) that users encounter when ordering products or services online.

Mistake four: you are not ready for an actionable service catalog

Many people have told us over the years that their organization simply was not ready to make the kind of transformation, from a technology- to service-focused organization, implicit in the mounting of an IT service catalog initiative. While it is true that a service catalog does force IT to think and act differently, our belief is that IT organizations no longer have a choice.

Change is being forced upon IT. There is nothing that an IT organization provides its business units and end users that they cannot buy from an outsourcer. In this sense, IT services are becoming a commodity of sorts. At the same time the expectations and awareness of an IT organization's customers are increasing day by day as they become increasingly technology savvy and technology dependant. The status quo is no longer an option.

Like a duck swimming in a pond, an effective IT service catalog can present a placid surface to the customers of an IT organization, while the feet paddle furiously behind the scene to keep up with new demands and process and organizational transformation. In this sense the service catalog can be used by an organization "not ready" to change as a means to buy time to be able to change.

CONCLUSIONS

Too often the relationship between IT and the business is characterized by mistrust. The cycle often starts when end users' expectations for some of the basic services IT offers are not met - a new PC is delivered later than expected, or setting up a new email account takes the user four phone calls to different help desks. When the business units ask "what's going

on?”, IT gets defensive and responds with five-nines uptime percentage charts and acronym-laden language that means little to anyone outside of IT. At this point, the cycle has already begun and trust is fading quickly.

At the heart of the problem is a breakdown in communication that impacts trust from multiple directions:

- End users don't know what to expect from IT, so they set their own expectations.
- Business unit executives don't understand or have the patience to navigate IT's siloed operations to get the services and innovation needed.
- IT communicates using terminology and context that is difficult for business unit executives and end users to understand.

Re-establishing trust between the business and IT means that IT must fundamentally change the way it views its relationship with business units and end users. Gone are the days when the internal IT organization owns a monopoly on service delivery. Now, the business demands transparency and visibility into the services and value delivered by IT. Today's business unit executives view IT operations like any other vendor - poorly executed service today means they may shift to another vendor tomorrow.

What is needed is a shift to a provider-to-customer relationship where IT continually markets the value of the services offered - where business unit executive and end user expectations are not only correctly set, but consistently met. A critical first step on this journey, in our opinion, is the creation of an actionable service catalog which defines and manages the relationship between IT and the business it serves.

Bill Fine (USA) leads product strategy at newScale, Inc., a service catalog software company. He is also the co-author of the “Pink Elephant Guide: Defining IT Success with the Service Catalog.”

Nick Schneider (USA) leads newScale's strategic consulting business and serves on the national advisory board for the itSMF USA. Nick holds an ITIL Masters Certificate and is currently focused on guiding organizations through their service portfolio journeys.

7.6 Request fulfillment - Ten best practices for managing IT service requests

What does it take to create an actionable service catalog and an efficient request fulfillment process? This article by Rodrigo Fernando Flores will outline ten best practices which can help your IT shop provide more responsive service delivery, increase operational efficiency, and improve customer satisfaction.

INTRODUCTION

Requests for IT services exceed the supply of available time and resources for most IT organizations. IT service teams in large corporations are constantly responding to requests from the business, often falling into the mode of reacting first to the end users who make the most noise. Meanwhile, employees complain that IT is difficult to work with, unresponsive, and takes too long to fulfill the services they need to do their job.

The explosion of application options, and the continually connected workforce have led to more and more pressure on IT organizations. Not only is there more technology per user, but the applications and options are more complicated than ever. As a result, the increase in the volume and complexity of service requests - requests for everything from access to applications, to software enhancements, to computer upgrades, to email and network access - have expanded dramatically.

That challenge is only compounded by the fact that each group within the IT organization usually has a different system in place for managing requests. And while IT is constantly busy, staff members are hindered by manual and inefficient processes - or working on tasks that may not be a top priority for the business.

There is a way to bring order to this chaos. It starts with the service catalog: defining a standard set of IT services, with associated service levels and costs, allowing users to order from this catalog of services, and then linking those requests to the request fulfillment process. By establishing standardized and repeatable services - and then publishing them in a self-service portal for IT consumers - the IT organization can provide a consistent way for users to request IT services and track the fulfillment for any request.

So what does it take to create an actionable service catalog and an efficient request fulfillment process? Based on dozens of proven examples with leading global IT organizations, this article will outline ten best practices to ensure success. By following these guidelines, your IT organization can provide more responsive service delivery, increase operational efficiency, and improve internal customer satisfaction.

But first, we'll cover some background on the typical IT environment and what the Information Technology Infrastructure Library (ITIL®) has to say about service requests.

THE CURRENT STATE OF IT SERVICE REQUESTS

Let's take a look at the typical IT request fulfillment environment from the perspective of the "requesters" and the "service delivery" teams responding to the request. One of the most immediate observations is that there are many different types of IT service requests.

These range from end user move-add-change requests (e.g., on-boarding a new employee, upgrading a laptop), to work requests for a business application (e.g., hosting an application for the marketing department, changing a report), or more technical requests where one IT group requests services from another IT group (e.g., deploying a new application environment, setting up a new server in the data center). Depending on the type and nature of the request, each is handled differently.

- **For requesters** - Most companies have multiple, conflicting and disparate mechanisms for ordering or requesting internal services on the front end. Does this sound familiar? Multiple intranets, web forms, Lotus Notes databases and custom applications all over the place; new forms getting developed all the time; employees and managers are forced to retype the same data again and again; end users continue to call the service desk or use the "shoulder tap" method to submit service requests; and business leaders are frustrated by their inability to order just once and then have IT deliver the service reliably.
- **For service delivery teams** - On the back end, most IT operations are organized around functional silos, characterized by their own set of processes, systems, and standards. Relatively few IT organizations offer a central catalog of standard service options appropriate for the user's business needs; even fewer have automated delivery of those services. Lack of coordination between service delivery teams and a reliance on multiple offline and online service request systems leads to service delays and inconsistent service quality.

This creates chaos and confusion for both the requester and the IT service delivery professional, the burden is on the requester (or designated "process shepherds" and coordinators) to navigate through these various ordering mechanisms, determine who does what, manage the various tasks associated with every service, and ensure that the end-to-end service is fulfilled. And the service delivery teams spend an inordinate amount of time responding to requests, researching requirements, validating information, and providing status updates.

To further illustrate what a mess this can create, consider these examples:

- **For end user services** - Let's look at the example of on-boarding a new employee. A computer must be ordered and configured with the necessary applications. Email and voicemail need to be set up, and the physical workspace must be ready and available. A hiring manager will typically need to consult the IT Desktop and Facilities teams, and possibly the Telecom or Network groups - and be the champion pushing to get these various service activities done on time and in the right sequence - through phone, email, and various paper or electronic forms. This takes up valuable time for the hiring manager, and it reinforces the perception that IT is difficult to deal with, that IT doesn't understand his/her needs or those of the business as a whole, and is simply too slow.
- **For IT-to-IT services** - Within large IT organizations, the application development group experiences similar frustration when they request infrastructure services (set up a server, provision storage capacity, etc.) from other groups within the IT organization. Lack of standards results in each request being treated as a one-off project, even if that same request is made dozens or hundreds of times. Without a standardized, repeatable

process, these services take longer than is required. And without a centralized way to track and prioritize each request, IT resources can be misapplied. These lapses can result in costly delays, both in hard dollars and in lost productivity.

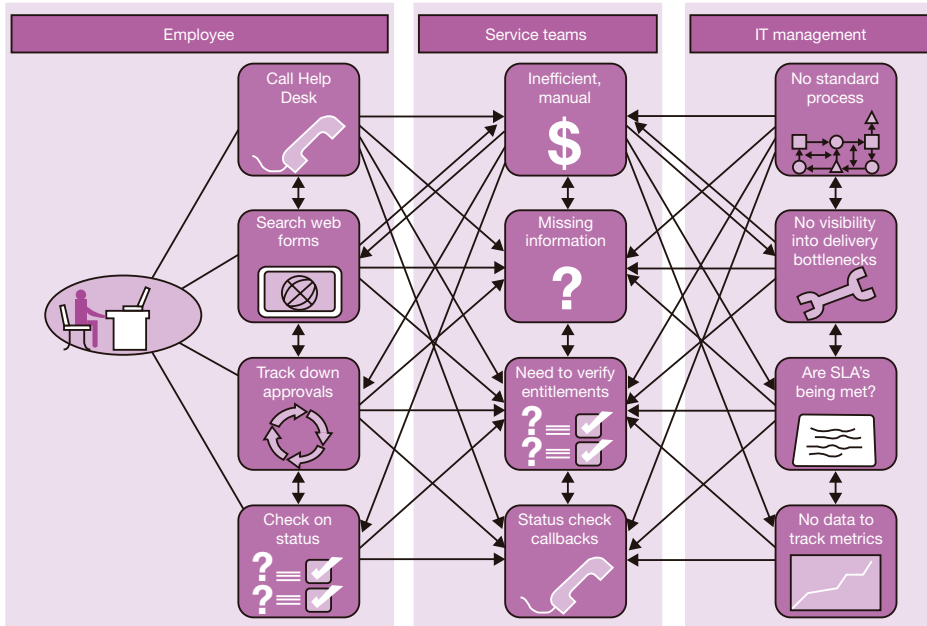


Figure 1 Typical service request process = complexity, confusion, complaints

Given the current state of request fulfillment within most IT organizations, this is clearly a challenge and a priority that must be addressed.

ITIL V3 - THE SERVICE CATALOG AND REQUEST FULFILLMENT

The ITIL framework is becoming the most common methodology deployed by IT organizations today, and the ITIL books and ITIL training organizations articulate the need for a service catalog. Unfortunately, prior versions of ITIL didn't provide much in the way of guidance on how to create a service catalog, or how to manage service requests. The ITIL definition of service requests is relatively limited (e.g., password resets) and they were covered within the incident management process, leading to much confusion among practitioners.

However, with the most recent refresh - ITIL V3 - there are new processes for both service catalog management and request fulfillment. Within ITIL V3, request fulfillment is introduced as a new process responsible for managing the lifecycle of all service requests. In the opinion of this author, it should have been called "request management" since there is a lot more to managing service requests than simply the fulfillment. However, it's a move in the right direction. And, for the purpose of this article, we'll use the ITIL V3 term of request fulfillment.

Within ITIL V3, here are some of the key requirements identified for request fulfillment:

- **Separate request fulfillment from incident and change** - Industry experts and ITIL practitioners have argued for years that it made no sense to have service requests as

part of incident management. For one thing, it messed up reporting because we want to reduce incidents - whereas service requests keep growing as the organization grows. It just makes sense to have a separate process for service requests.

- **Request fulfillment needs to be driven by standard services that are defined in the service catalog** - In large organizations, there are multiple organizational units, each requiring different service offerings and service agreements. The offerings defined in the service catalog should govern the request fulfillment process for each of the employees in those organizational units. This is a powerful concept and one of the most exciting new developments in ITIL V3.
- **Web self-service as a core function** - This seems obvious, but now it's recognized that a best practice for request fulfillment is to enable a user to go to a web self-service portal to search, compare, learn, decide, request, and track his or her request. This easy-to-use front-end interface can dramatically reduce calls to the service desk. In fact, requests may bypass the service desk by integrating directly with back-end fulfillment and provisioning tools.
- **The use of a "menu" of choices for the user** - One of the "critical success factors" (CSFs) for request fulfillment in ITIL V3 is the need to publish the service catalog as a readily-accessible menu of choices for the user to request services from IT. Essentially, just like a category-type-item structure is needed for incident management, this self-service menu is core to request fulfillment.
- **A single point of contact** - The service catalog and this self-service portal interface should be the single place for all service requests. If you have multiple channels for finding services and submitting requests, your users will be confused and adoption will be your single biggest challenge for request fulfillment.
- **The use of a "shopping basket"** - Your users are consumers and they will order services from IT just like they order other goods and services on the internet. A shopping basket is one function of an e-commerce interface. To create a positive "shopping" experience for your users, you also need to provide category management, search, service comparison, service guidance, status tracking, content management, account management and self-service configuration.
- **The need for authorization management** - Service requests are very different from incidents. For example, they often need to be authorized (i.e., who is authorized for what services) - whereas incidents typically do not. ITIL V3 finally recognizes the need for authorizations. These authorizations may include organizational, financial, security, and other authorization types.

So with V3, we finally have some official guidance from ITIL for managing service requests and streamlining the request fulfillment process. Whether you are embarking on an ITIL initiative or any other IT transformation initiative, an actionable service catalog coupled with effective request fulfillment is essential to your success.

By deploying your service catalog as a portal for the standardized services that IT offers, you can use this as the central point of entry for all IT requests. This actionable service catalog provides the primary means by which requesters interact with IT service delivery teams, and the primary interface between IT and the business.

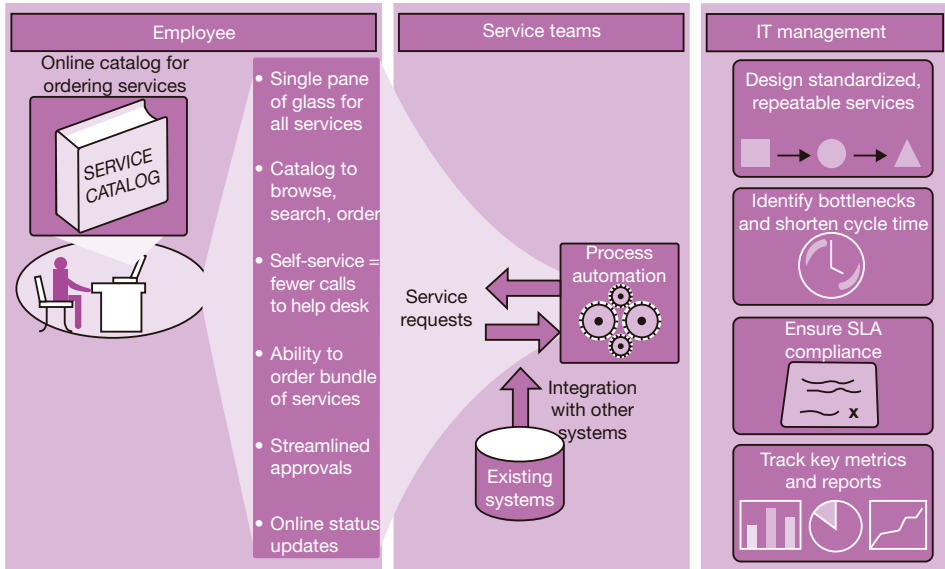


Figure 2 Best practice request fulfillment = consistency, reliability, efficiency

TEN PROVEN BEST PRACTICES FOR REQUEST FULFILLMENT SUCCESS

So what does it take to create an actionable service catalog for effective and efficient request fulfillment? Based on my experience with service catalog management and request fulfillment at dozens of leading Fortune 500 companies, here is a list of ten practical and proven best practices that can help to ensure your success.

1. Make your service catalog easy to find!

The service catalog is not effective unless your users actually use it for requesting services - otherwise, it becomes a static document and they will pick up the phone and continue to make expensive calls to the service desk. You need to design your service catalog with the objective of encouraging self-service adoption and usage from the outset.

When it comes to user adoption, you can apply the best practices of media and online commerce companies like Google, Yahoo, Amazon, and eBay. You need to create something compelling and unique to entice users to adopt the service catalog and move to self-service for request fulfillment.

The key is to implement what we call "YADU":

- Yahoo-like search and categorization
- Amazon-like merchandising
- Dell-like configuration
- UPS-like tracking for services

By following the YADU model, your IT organization will help to ensure adoption, alleviate challenges in finding services, and reduce your user training costs. IT's internal customers and users will intuitively understand how to navigate and use your service catalog because it's designed in a familiar model that they use every day as consumers.

First they need to find what they are looking for, and Yahoo provides a good example here. The Yahoo site became popular based on its categorization system, not just its search engine. Make sure your service catalog provides a systemic categorization, sub-categorization and cross-linking system so that it's easy to discover the services you offer. But don't forget the search engine.

In our experience, most first-pass implementations generate dozens of requestable service catalog items. For example: Install Windows XP, Grant Access to SAP Finance for this AP Clerk, Change this Report, Upgrade this Server... and many, many others. We have seen that a typical Fortune 500 company may have several hundred unique items in its IT service catalog within 12 months after the initial deployment. This complexity means that a categorization system and a search engine are imperative elements of the service catalog.

2. Merchandise, sell and cross-sell!

Your users are all familiar with e-commerce shopping sites: use that to your advantage. Market your services. Whether for installing a PC, granting access to a server, adding a report to an application, or setting up a new employee, all of these items should be in your service catalog.

Here are some recommendations for marketing these services:

- **Use merchandise “bundling” to remove complexity** - For example, a “New engineer setup” should include all the sub-services a new engineer needs to work, while “Move an employee” will require coordination across a variety of different tasks in the right order.
- **Don't skimp on content** - Explain the service and educate your consumer on exactly what they are getting. No one uses Amazon.com because of their part number system. It's their integrated content experience with reviews, comments, and graphics that provide a context to help people order. This also improves ordering, reduces returns (complaints), and eliminates rework.
- **Cross-sell** - Using the Dell.com model for PC configuration, let them configure their service and submit a request with the options and service levels that they are entitled to, based on well-defined standardized service components. While it may sound counter-intuitive, cross-selling and promotion techniques ensure more complete requests that decrease the need for clarification call-backs. By helping the user understand his or her choices and all the optional elements the first time, you can reduce the need for additional approvals and multiple dispatches.
- **IT services aren't free** - Define how your IT services are billed or allocated, and include that information in the service catalog. Let your users know that services come at a “price”, even if you don't use chargebacks. By simply assigning a price to the request, you can regulate the consumption of expensive IT services.

3. Help your customers help you!

End users don't usually complain that they have to fill out service request forms in the first place, but they certainly don't like having their request returned/rejected, or the service not delivered because the form wasn't filled out correctly. The second biggest gripe is that they need to fill out too many forms with the same information to request what they view as one service.

Look at it from their perspective and you will see that they think in “whole experiences”, very much like a customer ordering a “#1” at McDonald's. Here are some key suggestions for making request forms easy-to-use:

- **Use “English” in your form** - Service request forms are hard to understand because they use technical language with which the user is not familiar. Design forms that ask simple questions in common business and consumer-oriented language.
- **Design intelligent, interactive forms** - Use look-ups and pick lists, with integration into existing systems to minimize entry errors and user frustration. If you can look up live data to pre-fill the form fields, it's even better.
- **Re-use common elements** - Don't ask the user to fill out the same information multiple times. Create a common dictionary and content system. While this is an obvious practice in large-scale software development, service request forms often don't have common elements in a library that can be re-used.

4. People love UPS tracking! Use it to reduce service desk calls by 30 - 50%

The single biggest frustration for your users is that they don't know what is going on, when their request is going to be fulfilled, and what needs to happen to get their service.

Deal with this problem and people will think you are a service star (and reduce calls to the service desk by 30-50%). Not being able to predict delivery timeframes and not knowing the status causes frustration and creates direct economic costs in the form of back-and-forth status calls, over-ordering, emergency expenditures and maverick service requests. Without clear dates, commitments, and valid status, end users are forced to wait, impacting their productivity.

Here are some tips for status tracking:

- **Provide visibility for all processes** - Like UPS, you should provide insight throughout the service request cycle. This means you need visibility to your authorization process, your delivery process, your outsourced processes, and your cross-system processes. This is achievable today by using service catalog software with a modern enterprise integration framework.
- **Use rich emails** - People love to be in control. The right email, at the right moment, with the proper information, formatted professionally, let's them know you are in control, and their request is under control.
- **Provide a “receipt” not just a tracking number** - A web page for ordering services is not sufficient if the customer has no meaningful record of their request, the services ordered and the dates expected.

5. Under promise and over deliver! Manage expectations to reduce firefighting

Satisfaction and quality are the assessments that people make between their expectations for a service and their perception of the service performance. You can be satisfied with a five dollar burger at McDonalds, while another might be dissatisfied with a hundred dollar meal at a four star restaurant - different expectations lead to different perceptions. Typically for IT service organizations, the expectations for a service are not set appropriately.

To effectively manage expectations:

- **Be explicit on dates** - This is tough for IT service delivery, so establish upfront three key dates you will track: expected date, estimated date and actual date. These are your key dates for managing the requests.
- **Own the promise for the whole of the experience** - The customer judges their satisfaction based on the whole package, not from a task or function. Ensure that you are clear about what it takes to fulfill the entire service request process, not just the initial task.

- **Close the loop** - Don't just complete tasks, inform the user of the status and ask for an evaluation. Make sure the communication links back to the requisition and the service catalog so the user can clearly navigate from notification to requisition.

6. Make it fun! An educated customer self-regulates behavior

Fun is an important part of the service experience. People pay more and are more satisfied when they enjoy themselves. Seems obvious, no? But applying this to IT and services seems counter-intuitive and difficult to sell to top management. Here's why it should be considered: you will increase use of the service catalog, you will reduce the number of complaints, you will reduce the number of fires and emergencies, and you will reduce rework.

Remember, customer satisfaction is set by reliability, but also by empathy, understanding, courtesy, credibility and competence. Here are ways to make "fun" a serious endeavor, by inserting fun into IT service requests:

- **Let them see how you "manufacture" the service** - Letting your requestors understand the "manufacturing" tasks makes them more knowledgeable and sympathetic with your delivery process. This can be accomplished by giving them a view of the workflow tasks involved in their unique request.
- **Give them choices** - People love shopping and having choices. So why not give them meaningful options for their services? Examples of simple things that don't cost a thing include the ability to schedule a convenient time for a move or having multiple choices for a service. If you use the right service catalog software, this is easy and cost-effective. The result: people take much more responsibility for their choices.
- **Give your service personnel autonomy** - Having autonomy is not contradictory to having processes and standards. In fact, by having standard processes in place, you should be able to grant more autonomy to service personnel. This prevents centralized mistakes from spreading to every facet of your business.

7. Stuff happens! Make it easy to deal with breakdowns and exceptions

Regardless of how well you do, or intend to do, breakdowns happen. Sometimes those are easy enough to correct, other times they are not.

- **Focus on service quality** - Would any manufacturing company make a product without having quality assurance, return, support and credit processes? No, and neither can a service organization. Master the three key processes for service delivery:
 - the process by which you "manufacture" the service itself
 - the process by which you deal with breakdowns in the delivery of the service
 - the process by which you redress or "make good" on any failure
- **Allow for exception requests** - Regardless of how good your service catalog is, how complete your catalog is, or how thorough your order forms are, they can't account for everything a user may request.

8. Let them know how great you are doing! Provide a service dashboard

Providing dashboards for business managers is the best way to create a shared understanding and ownership for IT services. By using honest metrics that matter to these stakeholders - e.g., how often does the manufacturing group request a particular IT service - they become partners in service delivery. The key is to replace anecdotal word-of-mouth with factual metrics available at their fingertips. Here are some recommendations for incorporating a dashboard into your service catalog:

- **Publish meaningful metrics** - Metrics should be in the language of the customer, and they should be relevant to the behavior. Metrics such as system uptime, call volumes, etc.

are not particularly meaningful to business stakeholders. Metrics such as how frequently you delivered on the promised delivery time for on-boarding a new employee are very meaningful.

- **Let them see the behavior of their team** - What are the top ten service requests? What is the average approval time for a service request? What costs are they incurring? This data is easy to get by using an actionable service catalog, yet today it is often impossible.
- **Market your performance** - Again, the perception and expectation of service quality are tied together. Set simple metrics in your service level agreements (e.g., the estimated time to deliver each service), track your performance against those metrics, and publish those results in the service catalog portal. In doing so, you can create the perception - and the reality - of reliability.
- **Dashboards should be short and to the point** - Don't overwhelm your stakeholders with lots of useless data, if they ask for more, great! But keep it short and simple. Think of it as publishing a short newspaper on a daily basis.

9. Don't put lipstick on a pig! Take care of the back-end integration

There's no point in putting up a service catalog that is not integrated with the back-end request fulfillment process, or that doesn't provide and track real estimates for service delivery. The most common mistake is to create a service request form that simply generates an email to someone, or an online service catalog that simply documents the service description and provides the email contact information. This often fails because it ends up being a constant source of maintenance (hence costly) and it doesn't help manage either customer expectations or service delivery.

Some tips for avoiding this mistake:

- **Ensure seamless integration with the back-end** - The service catalog should be able to integrate with the myriad of different back-end systems in your company, including your directory, enterprise portal, service desk system, procurement system, asset database, and configuration management database (CMDB). This can typically be addressed by leveraging packaged service catalog and request management software with a robust integration framework and pre-built adapters.
- **Develop integrated service plans** - Bundling is a powerful concept for simplifying the request management experience. The back-end construct of bundling requires a workflow to manage the "manufacturing" of the service tasks spanning people, departments and systems. This removes the need for personnel to manually shepherd requests - these savings alone provide the hard dollar ROI to justify the entire service catalog deployment.
- **Integrate with non-ITIL processes** - In my experience, there are a number of non-ITIL processes involved in request fulfillment. For example, a service request may invoke the procurement process or tasks to be fulfilled by departments outside of IT, such as facilities and human resources, or the service may be fulfilled by external suppliers, outsourcers, and managed service providers. Your service catalog and request fulfillment process should coordinate these various interactions and tasks - whether they are defined in ITIL or not.
- **Manage the end-to-end service** - Roll up estimates for delivery from multiple systems and service providers in real time, and communicate those estimates through your service catalog. This allows you to manage customer expectations and notify them of changes.

10. Keep it fresh! Make the service catalog easy to update by the service owners

Just as your IT services change, so will your service catalog content. In our experience, we find that the number of IT service request variations (including moves, adds, and changes) typically ranges from one hundred to several hundred. Over time, the range of IT services and the volume of service requests will continue to grow. Additionally, service expectations change over time, so there will always be variations driven by technology change, competitors, and economic or regulatory volatility.

Managing content in your service catalog is critical:

- **Make it easy to update** - Provide tools that allow service managers or service owners to keep their services fresh. Change will happen, and you don't want bottlenecks to keep your service catalog from being fresh. Flexible and codeless (i.e., no programming required) service design tools are extremely important for keeping the total cost of ownership under control.
- **Have well-defined roles for maintaining services** - Your service catalog strategy should include clear roles and functions for managing large scale changes to services and the fulfillment process. Most organizations have different people updating different aspects of the service catalog (e.g., updating the images and descriptions versus changing the fulfillment workflow); distributed role-based access to carry out these updates is an important aspect of any service catalog initiative.

CONCLUSIONS

There is a way to bring order to the typical chaos of managing and fulfilling IT service requests in a large organization. By following the ten practical tips outlined in this article, you can deliver a relatively quick and easy win for IT:

1. Make your service catalog easy to find!
2. Merchandise, sell and cross-sell!
3. Help your customers help you!
4. People love UPS tracking! Use it to reduce help calls by 30 - 50%.
5. Under promise and over deliver! Manage expectations to reduce fire-fighting.
6. Make it fun! An educated customer self-regulates behavior.
7. Stuff happens! Make it easy to deal with breakdowns and exceptions.
8. Let them know how great you are doing! Provide a service dashboard.
9. Don't put lipstick on a pig! Take care of the back-end integration.
10. Keep it fresh! Make the service catalog easy to update by the service owners.

Whether you are implementing ITIL V3 or if you just want to provide better service to your users, the key to effective request fulfillment is your service catalog. By providing an actionable service catalog linked to your request fulfillment process, you can improve service to your users and enhance IT's reputation. And by streamlining the delivery of services, you can achieve a very attractive return on investment.

In closing, doing anything according to "best practice" guidelines is always difficult. But perfection is the enemy of action, so use this set of best practices as if it were a map for a vacation. It shows you all the interesting places you can go, but you don't have to go to all of them immediately. You can start with just a few services in your service catalog and then expand over time.

Just get going now and use these ten proven best practices as your practical guide to deploying an actionable IT service catalog for request fulfillment.

Rodrigo Fernando Flores (USA) is the founder and chief technology officer of newScale, Inc. Rodrigo is an active member of the itSMF USA and is an internationally-recognized speaker with expertise in service catalog and request fulfillment best practices. He has advised dozens of IT organizations around the world, and has led service catalog training courses across the U.S. and Europe for hundreds of IT executives.

7.7 The unclear relationship between change, release and project management

Do you have a clear picture of the interfaces and relationships between change, release and project management? Even after analyzing what ITIL and PRINCE2 - market-leader frameworks in the process domain - suggest? In this article, Christian Cantù and Maxime Sottini draw a picture of the potential relationships between these processes, and give practical guidance on how to implement them.

INTRODUCTION AND OBJECTIVES

Service and project management are recognized to be among the core disciplines in order to effectively and efficiently manage IT organizations and achieve IT goals. Project management is a mature discipline, evolved outside the IT domain, but widely adopted in IT too. The modern development of project management starts in the fifties, although Gantt charts were developed and used before this. From its origin, different approaches and frameworks have evolved and emerged. Among them, some have been more successful, for example PMBOK® (Project Management Body of Knowledge) and PRINCE2™ (Projects IN Controlled Environments). The differences between these frameworks are not so relevant for the purposes of this article and references from the latter will be used where necessary to aid comprehension and limit the need for a glossary and explanations.

Service management has developed later in comparison to project management. In the IT domain, ITIL® (the IT Infrastructure Library), has greatly contributed to the development of the discipline from the eighties to present day.

Both project and service management have specific processes as central components of the respective disciplines. Applied to the same IT management domain, these processes are used to support the same items, such as applications, infrastructures, resources and objectives. Although in the specific frameworks there are some attempts to clarify the general relationships between the different disciplines, many practical questions arise when implementing these disciplines and focusing upon the detail. Theory is not able to answer some questions, often related to the joint use of the processes suggested by the two disciplines. But there are also aspects relating to organization and tools that require further understanding.

The main area of improvement in terms of understanding and knowledge lies in the relationship between project and service management. In the IT domain, the main objective of project management is to support a controlled evolution and transformation of applications and infrastructures. Service management supports the effective delivery of applications and infrastructures. At first sight these two aims appear to be synergistic, but on looking closer we can easily find overlapping areas. This is because IT services are not static, rather they evolve continuously and therefore IT management is concerned about the evolution and transformation of services. Before the advent of ITIL V3, the main processes able to guarantee this evolution were release and change management. These two processes have

a strong relationship with project management processes. In some cases they are mutually exclusive, in some others strongly synergistic. Small service evolutions, for example, may be managed without project management techniques and processes, instead using the basic service management processes such as change management. On the contrary, a significant service evolution will probably need to be managed as a project, which may include the management of several releases.

When considering these relationships, which whilst they are clearly evident are certainly not the only such ones, it is clear that some practical guidance on how to define and apply them would be useful. When does a change have to be managed as a project? Does a release have to be managed as a project? When does a project activate changes and releases? Which are the correct responsibilities and activities of project and change management functions? These are only some of the many questions arising when considering project and service management. The aim of this article is to explore them and to supply practical guidance alongside the theoretical approaches, with explicit references to ITIL and PRINCE2.

In order to achieve this aim, the following steps will be performed. A brief introduction of service (ITIL) and project (PRINCE2) management theory will be given. This introduction will support the identification of change and release management as the main processes to focus on, and will give a better understanding of the main issues concerning the relationships between the target processes. These issues will be explored in a dedicated section. Later, a general overview of the relationships between the processes, including a graphical model, will be given. This will be used to identify the specific interfaces between the processes, and these will be explored in further details, by means of the following specific sections: change and release, change and project, and project and release. Finally, a number of conclusions will be drawn.

ITIL THEORY OF CHANGE AND RELEASE MANAGEMENT

Change management is often related to people (a structured approach to manage the impact of change to individuals, teams, organizations), but in our context we mean a specific process of IT service management. This process has been well described in ITIL V2:

“Change management is responsible for controlling the Lifecycle of all Changes. The primary objective of Change Management is to enable beneficial Changes to be made, with minimum disruption to IT Services”, where Changes are meant as “The addition, modification or removal of anything that could have an effect on IT Services. The Scope should include all Configuration Items, Processes, Documentation etc.”

In ITIL V3, the process has been placed into the Transition phase of the Service Lifecycle with minor modifications to its contents, though this is not relevant to the aim of this article. Change management is a sequence of activities starting from recording the Request for Change (RFC) and finishing with change review and closure, through the steps of assessment and authorization, planning and implementation, in order to obtain the intended modifications, subject of the initial RFC. The process according to ITIL V3 is described in figure 1.

Change management is normally performed by permanent service management roles and functions of the organization, such as the change manager, the Change Advisory Board (CAB), and service owners; it is supported by service management integrated tools.

The unclear relationship between change, release and project management

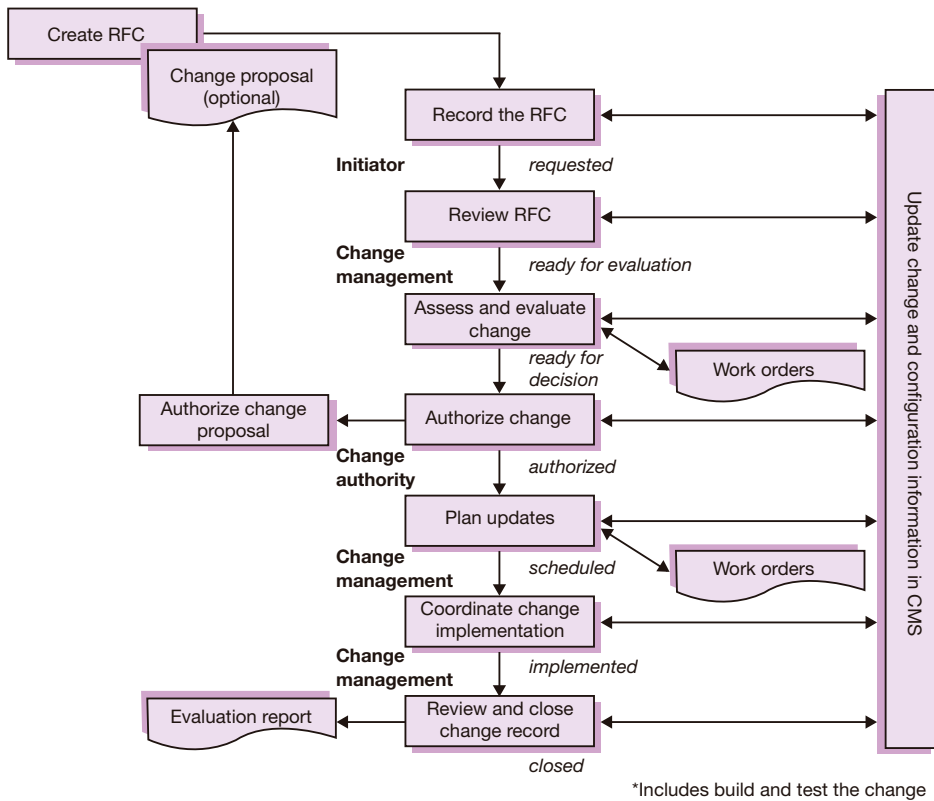


Figure 1 The change management process according to ITIL V3 (Source: OGC)

In ITIL V3, some relationships between project and change management are identified. Probably, the most important is defined in Service Design, where in the principles it is stated that the introduction of, or the major modifications to, IT services have to be managed by means of projects. Figure 2 depicts how the project approach is needed to manage the stages within the lifecycle for the deployment of a new or modified service (change). Figure 2 also highlights the role, among all the Service Lifecycle processes, taken on by build, test, release, deployment management and change management.

In ITIL V3, specifically in the change management process description, some examples of change request are listed, together with the procedures to be used to manage them. We can derive the following:

- Projects should submit changes affecting services to the service change management procedure.
- Projects should use standard changes and service requests for pre-authorized typical operational requests (e.g. user access request).
- Projects have specific change management procedures for changes not impacting the service or design baseline.

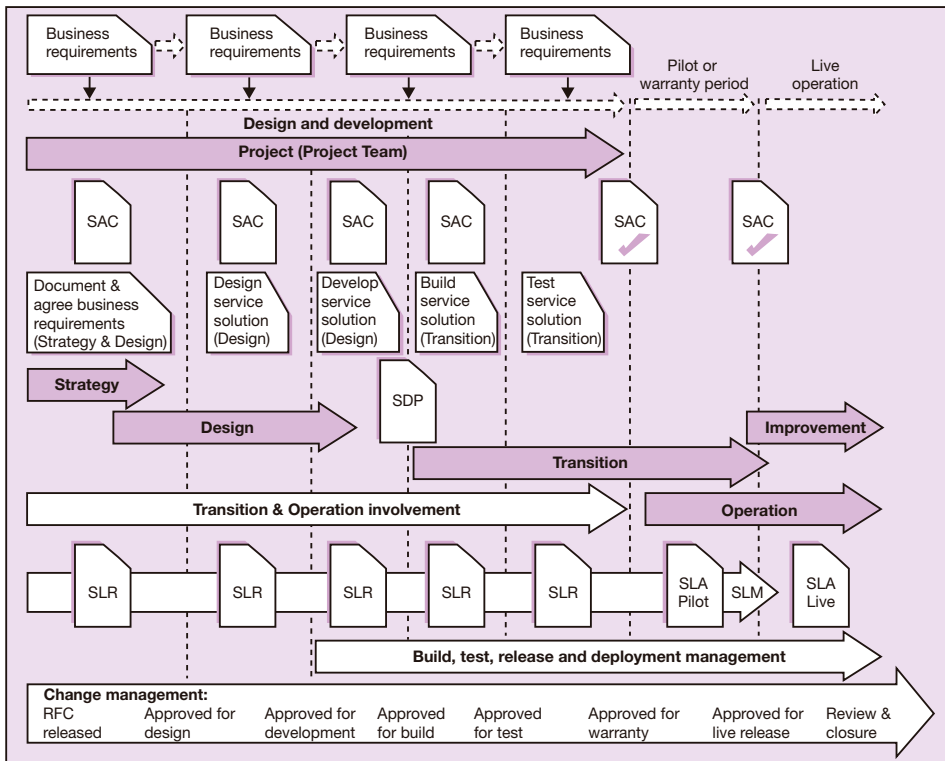


Figure 2 Relationships between project management and service lifecycle phases (Source: OGC)

Release management has greatly evolved with the new version of ITIL. The definition of the process in version 2 is the following:

“The Process responsible for Planning, scheduling and controlling the movement of Releases to Test and Live Environments. The primary objective of Release Management is to ensure that the integrity of the Live Environment is protected and that the correct Components are released. Release Management works closely with Configuration Management and Change Management.”

In the ITIL V3 definition, release management has become *“part of the Release and Deployment Management Process”*. For the purpose of this article, we will consider the definition contained in ITIL V2 to be more appropriate, thus viewing release management as being made up of planning, building, testing and deploying activities. Within this scope we will consider the relationships with the other processes, change and project management.

One of the clearest relationships between release and change management is depicted in figure 2. Release management is the process that actually takes changes into the live environment. From the state “Approved for build” onwards, the change is executed through release management. In other words, each release may include and implement one or more changes.

Figure 3 depicts another existing interface between release and change management, a relation according to which an RFC is created in order to activate an instance of the release management process; this, in turn, deploys a specific release and, therefore, brings specific changes into a live environment. This interface also highlights the role of change management as a means to execute some steps of the release management process.

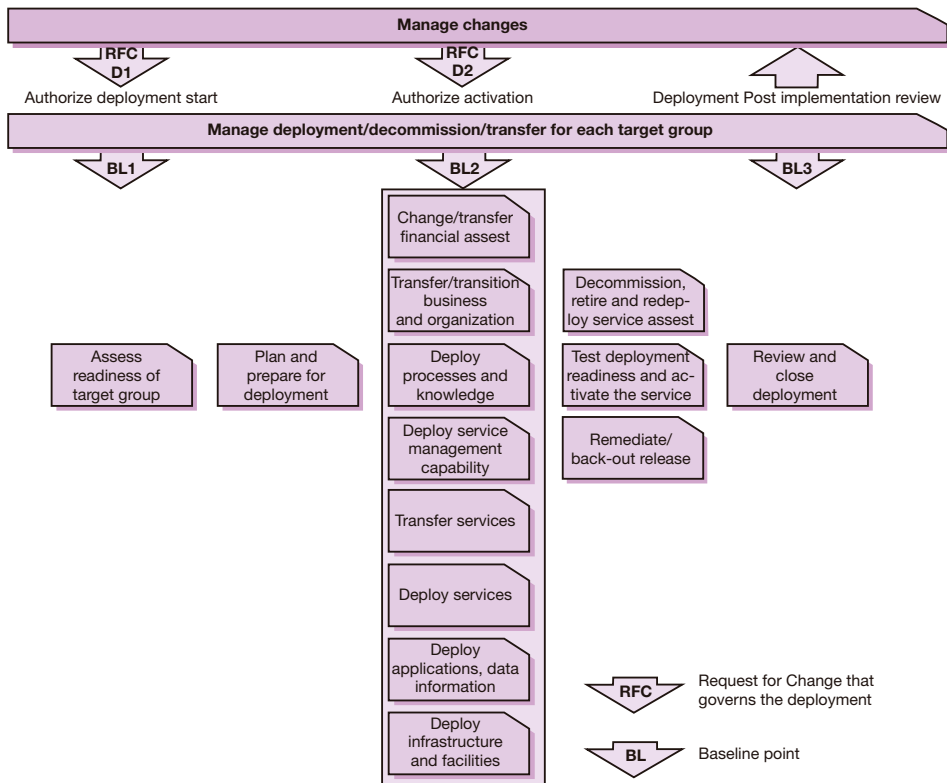


Figure 3 Some relationships between change and release management (Source: OGC)

In synthesis, the main interfaces identified between change and release management are as follows:

- Releases may embed one or more changes which are deployed and finally implemented through those releases.
- Some activities of the release management process, such as installation of builds, may be activated and performed by means of changes.

Finally, and regarding the interfaces between project and release management, in ITIL V3 an explicit relationship is identified in planning activities. Project plans should include release activities as well as the resources needed to perform them. Among these activities, as an example, an important role is played by testing. More generally speaking, the release management approach, e.g. the V-Model referred to by ITIL V3, shall influence project plans definition.

PRINCE2 THEORY OF PROJECT MANAGEMENT

Founded in 1975 and originally known as PROMPT, this project management framework was adopted by the Central Computer and Telecommunications Agency (CCTA, then Office of Government Commerce, the same founder of the ITIL framework) as a standard to manage information systems projects. It was developed and refined over years of use till the current version, PRINCE2™ project management method.

PRINCE2 is a repeatable method, built on experience, available to organizations willing to adopt a structured “project” approach to deal with the creation and delivery of new business products, or to implement any change to their environments. It is therefore not only applicable in the IT domain.

A key principle of the method is its “product-based” approach, i.e. an approach focused on the products (deliverables) produced by the project along its lifecycle, and not the activities performed to produce those products. The core structure of the method is made up of:

- processes, that provide a controlled start, progress and closure of any project
- components, i.e. the different aspects of a project that are taken into consideration (e.g. business case, organization, controls, plans, ...)
- techniques, supporting the implementation of processes

Figure 4 depicts the process model as detailed in the PRINCE2 framework, a model that needs to be tailored to the specific requirements of the project/organization.

The model is made up of eight processes:

- “Directing a Project (DP)” is the process aimed at the senior management team (project board), responsible for the whole project; the key decision-makers are, therefore, specifically involved in this step, that includes approval of the project, monitoring of its progress and closure.
- “Starting Up (SU)” and “Initiating a Project (IP)” are two initial steps aimed at assessing all the information and knowledge of the project, such as the project aim, project management team, customer’s expectations, and establishing a baseline against which measure the progress. At the end of these steps, a final confirmation for project activation and a plan for the whole project are obtained.
- “Controlling a Stage (CS)” is one of the core process steps owned by the project manager, who is involved in monitoring and controlling activities, in order to manage a single stage from approval to completion.
- “Managing Product Delivery (MP)” provides a control mechanism to ensure the production and delivery of project’ deliverables (products).
- “Managing Stage Boundaries (SB)” allows the project board to be aware of the status of project plan against its business case and the project to go ahead, from one stage to the other.
- “Closing a Project (CP)” is the step according to which the project manager asks the project board for authorization for project closure.
- “Planning (PL)” is the repeatable process step used whenever a plan is required or needs to be updated.

PRINCE2 also includes components and techniques which add guidelines and hints to manage projects successfully. Components are:

- Business case
- Organization
- Planning (this is also a process)

The unclear relationship between change, release and project management

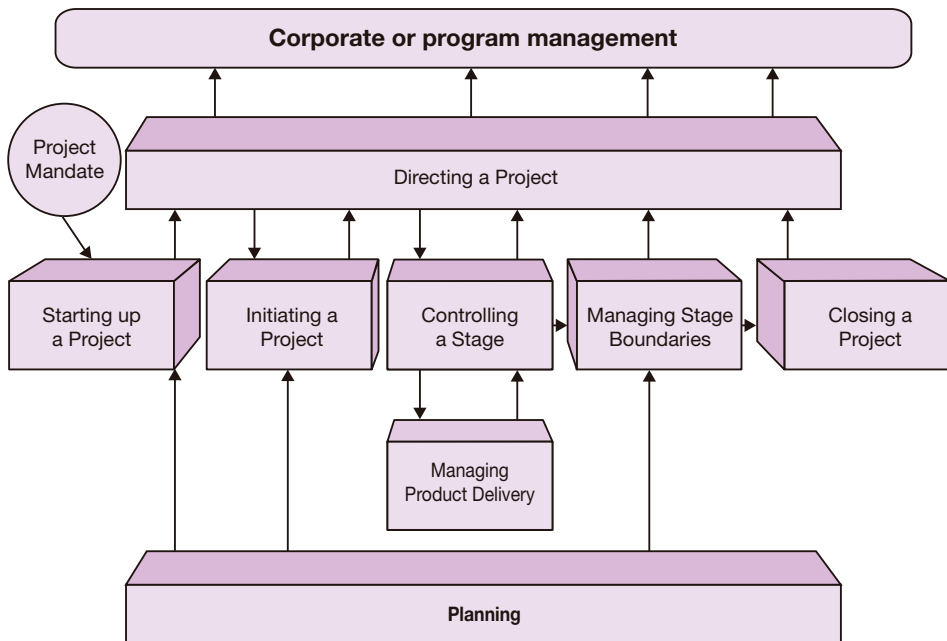


Figure 4 Processes in PRINCE2 (Source: OGC)

- Controls
- Management of risk
- Quality in a project environment
- Configuration management
- Change control

The final component seems to be apparently related to service change management, but similarities end as soon as you begin to get into detail. Change control deals with changes to the scope and specifications of a project and establishes a formal authorisation process for them. A request for change is a request to alter the current specification of a product.

Given the focus of this article, which is examining the relationship between change and release management processes, there are several concepts of interest in PRINCE2:

- Product orientation - Products are deliverables of the project and they are derived as outputs of service management changes and/or releases.
- Stages - These are phases of the project, each requiring formal approval to commence and to be closed. They may correspond to the phases of a release (for example deployment).
- Project board - This is the decision authority of the project, approving initial objectives, scope, products, etc., and any deviation to it, including changes as previously mentioned and intended.
- Work packages - These are set of instructions issued to a team manager in order to complete an area of work for the project. They contribute to the implementation of products and, therefore, they may also contribute to the implementation of changes and/or releases.

If we look at work packages in particular, they are created during “CS1 Authorising a Work Package” by the project manager with input from the team manager and team members. Once created, the work package is accepted and agreed in “MP1 Accepting a Work Package”, where dates and other details are negotiated. A Work Package is later implemented through “MP2 Executing a Work Package” and finally delivered through “MP3 Delivering a Work Package” process.

Even if PRINCE2 theory doesn't highlight relationships between project management and change or release management explicitly, the main areas where these processes will be identified, are the CS (Controlling a Stage) and MP (Managing a Product) processes.

ISSUES REGARDING RELATIONSHIPS BETWEEN CHANGE, PROJECT AND RELEASE MANAGEMENT

Unless IT service management deals with day-to-day activities, it appears to be a separate and independent domain, unrelated to project management. But when service management is about important modifications to services, the borders are no longer sharply defined. In the previous sections, we have explored the target processes and the relationships identified in the theoretical approaches between them. In this section, we illustrate why further analysis and understanding is needed.

As discussed, some relationships have already been drawn from a theoretical point of view, but are they exhaustive? Some important relationships have not been identified yet. For example, do changes have to be managed as projects? Experience suggests that, in certain cases, when changes are significant in terms of their size and the resources required, then it seems to be reasonable, if not actually recommended. In fact, the change management process may be not suitable and sufficient to guarantee the control of risks and the achievement of all intended results. But when does a change have to be managed as a project? Is there a specific size of the change, according to which project management processes have to be activated? Is a PRINCE2 approach suitable to manage it, or is it too complex?

A similar question may arise for release management. If a release is not derived from an existing project, should it be still managed as a project? This situation is not depicted in theory, but in real life there may be complex circumstances that could lead to such an instance. Again, it is not obvious as to the conditions in which project management should be activated and which approach is best suited.

Other relationships have been identified, but not deeply analyzed. For example, the relationship between release and change management. Release management requires the installation of a specific release in different environments (system test, acceptance test, production, etc...). Should this task be managed through changes, should these changes be standard ones? After all, these are modifications of target environments. The theory does not give a clear answer.

Another relationship, not completely explored and whose impact hasn't been well addressed, is the one related to the organizational issues existing within change, release and project management. These processes may lead to different, and in some cases even contrasting, objectives: project management focuses upon deadlines, whilst with change and release management the emphasis is on assuring minimal disruption of existing IT services. Should

change and release management be supported by permanent functions or should the tasks and responsibilities be performed by transient ones, included in the organizational structure of the project? How should project management processes (in particular those responsible for delivering products, e.g. CS and MP) be combined with change and release management processes? These choices may reinforce or reduce the capacity to achieve specific types of objectives such as timing vs. quality.

Finally tools. Service (including change and release) and project management are mature markets for tools. But generally, these tools do not cover both disciplines. Some questions arise: what are the main tasks to be accomplished by each tool? Which are the main interfaces to be implemented to bridge them?

These are only some of the questions that this article will try to address in the following sections, each of which will focus upon two processes, illustrating and analyzing bi-directionally the interfaces between them. But first of all we will provide a general overview of all relationships.

OVERVIEW OF THE RELATIONSHIPS

We have identified two main scenarios to be considered in order to explore the relationships between change, release and project management processes. The first one is depicted in figure 5 and it is referred to as the day-to-day operations scenario. Service management processes take care of routine. Changes are required by problem or incident management as well as other processes, such as service level or demand management, and they can be related to small evolutions/enhancements/ fixings of the services. In this scenario, the interfaces to explore are:

- **Changes to releases (1)** - Changes are deployed by means of a release.
- **Release to changes (2)** - Specific changes are issued in the release management process in order to install and activate a release.
- **Release to project (3)** - A release (for example a major release) needs additional

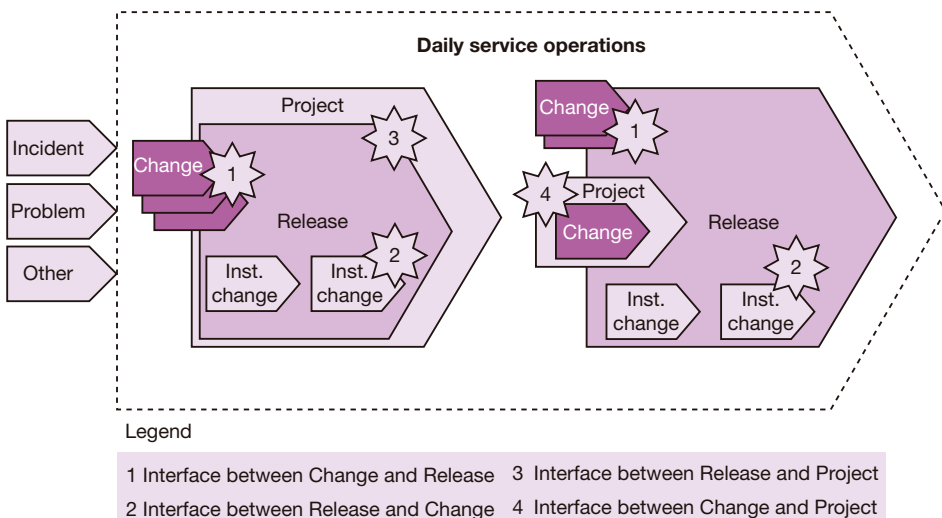


Figure 5 Relationships in the daily operations scenario

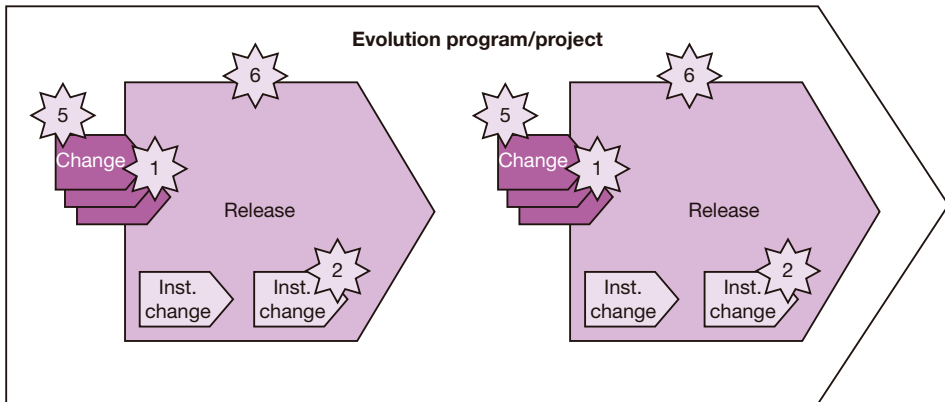
coordination and care in order to assure proper implementation by means of a dedicated project.

- **Change to project (4)** - A change (for example a major change) needs additional coordination and care in order to assure proper implementation by means of a dedicated project.

In addition, another scenario has been identified where relationships between processes may be activated. This is the scenario depicted in figure 6 where programs or projects will develop, and will implement business or IT requisites with impacts such as new or modified services. In this case, the program/project will interact with the service management processes, including change and release management. The interfaces to explore are therefore:

- **Changes to releases (1)** - Changes are deployed by means of a release.
- **Release to changes (2)** - Specific changes are issued in the release management process in order to install and activate a release.
- **Project to changes (5)** - Projects issue changes in order to implement specific requisites.
- **Project to releases (6)** - Projects cluster changes into releases, using release management techniques and principles to deploy and make IT deliverables operational.

Relationships 1 and 2 are present in both scenarios and it should also be noted that scenarios may coexist. Effectively, it is usual to have scenario one active for daily changes and the management of minor releases, and scenario two for major evolutions, managed as projects.



Legend

5 Interface between Project and Change	1 Interface between Change and Release
6 Interface between Project and Release	2 Interface between Release and Change

Figure 6 Relationships in the context of a program/project scenario

In the following sections we will analyze the 6 interfaces in more details.

EXPLORING INTERFACES: CHANGE AND RELEASE MANAGEMENT

In the overview of relationships, we have identified two interfaces between change and release management. The first one, interface 1 in figure 5 and figure 6, is described in ITIL:

changes are included in releases and deployed accordingly. Let's analyze in more detail the lifecycle of a change, and its relationship with release (see also figure 2). Changes are approved according to change management approval steps. The following steps, "design and development of the change", are performed through change and/or project management activities (this normally include some level of testing, typically regarding each single change). Some types of changes, for example configurations of software or substitution of hardware, may be completed through change management process steps. Others, typically those affecting applications, are better tested and implemented together, bundling them in a release and using the release management process principles and activities to complete the changes. The steps to be performed are, therefore, indicated by release management. For example, the service V-Model can be used to define the levels of testing: service release package test, service operational readiness test, service acceptance test. The bundled changes are synchronized with release statuses so, for example, when the release passes the operational readiness test, the changes included are commonly "Approved for Warranty". After deployment, see interface 2 described in the following paragraph, the "Review and Close Deployment" step of release management also completes the "Review and Closure" step for the included changes.

If a release is a means to deploy changes, then some steps of release management may profit from the change management process to be performed. This is interface 2, in figure 5 and 6. A release needs to be installed in several environments during the release management process. For each installation an RFC can be used for the target environment. A standard change is well suited to manage this type of request where the approval phase of the change can be linked to specific service acceptance criteria depending on the target environment/test phase to be executed and the type of release (minor, major or emergency). Also, the approval authority may differ, according to these drivers. The execution of these changes practically implements some of the activities of the release management process, namely those related to the Deployment (applications, data, information and infrastructure), as depicted in figure 3. Also remediation actions, such as the back-out of a release, can be performed inside an installation change (for example if an application is not running properly after installation) or by means of explicitly dedicated change (for example the installation of a previous version if test results are not satisfactory).

Table 1 depicts some possible correspondences between a specific type of release to be installed in a specific target environment and the related type of change to be utilized. Major releases will contain major evolutions/transformations but also bug fixes. Minor releases will include smaller evolutions and bug fixes too. Evolutions and fixes will be managed by means of changes included in the release for implementation and deployment. Installation changes are dependent on impact and urgency of the release but also on the target environment. For recurrent installations (such as weekly minor releases, with bug fixes and small evolutions) specific standard change procedures may be defined.

Change and release management may be implemented with different tools. The change management tool is normally part of the service management solution, while some release management functionalities may be supported by the software lifecycle management solution, including deployment (software distribution features). In the case of separate tools, it is important to implement the following interfaces in order to facilitate seamless and efficient processes:

- visibility of changes, as well as of their attributes and details, to be included in the release with the possibility to link them to the release record

- automatic update of the status of changes included in a release, based on the status of the latter
- automatic creation of the installation changes linking all the release information and its included changes

Release	Target environment	Installation change
MAJOR	Acceptance test	SIGNIFICANT
	Production	MAJOR
MINOR (weekly scheduled)	Acceptance test	STANDARD CHANGE
	Production	STANDARD CHANGE
MINOR (not scheduled)	Acceptance test	MINOR
	Production	SIGNIFICANT
EMERGENCY	Production	URGENT

Table 1 Sample of correspondences between releases and installation changes

EXPLORING INTERFACES: CHANGE AND PROJECT MANAGEMENT

The first interface we are going to explore is number 4 in figure 5, the interface between change and project. Some changes may be relevant in term of associated risks, costs and/or work effort. In such a case, simple change management process activities (see figure 1) may be not sufficient to guarantee the expected results. Since a change is a transformation, project management processes and techniques may be used to achieve this. There are two practical ways, which may coexist, to define when a change should be managed as a project: use of thresholds and human judgement. One of the most common thresholds to be used is the effort or equivalent cost (estimating all resources, included company's employees by means of standard or actual daily or hourly costs). An example of how to proceed using this type of threshold is depicted in figure 7.

The start of the diagram in figure 7 is normally the domain of the change management process, but having more than a single entry point for requests is probably unavoidable in medium to large organizations. The project management office, change management and demand management are usually the functions that will process the majority of evaluations and, therefore, they should be aligned as far as the drivers and thresholds are concerned, interacting and communicating as required in order to agree on doubtful cases.

The sample threshold used in figure 7 to define how to manage the RFC is forty man days, which corresponds to many actual situations we have encountered. These forty days can refer to either the internal or external resources needed to implement the initial request, with reference to the organization where the process takes place. This is not the only choice to be made as far as the process is concerned. Of course, a complete PRINCE2 structured process is too heavy for a 60 man days requirement and it is wise to consider other different project management approaches, with a level of structure adequate for the complexity and weight of the objectives. In figure 7, two possible models have been identified: the standard project model corresponds to the full PRINCE2 model (see figure 4) and is applicable to large and complex tasks; the light project model is a simplified process (see an example in figure 8, a simplified project management model based on PRINCE2 for single stage projects¹).

¹ It is not the aim of this article to fully explore the simplified model. Selected PRINCE2 processes shall be customized in order to provide a simple Project management framework, suitable for single stage small to mid-sized Projects.

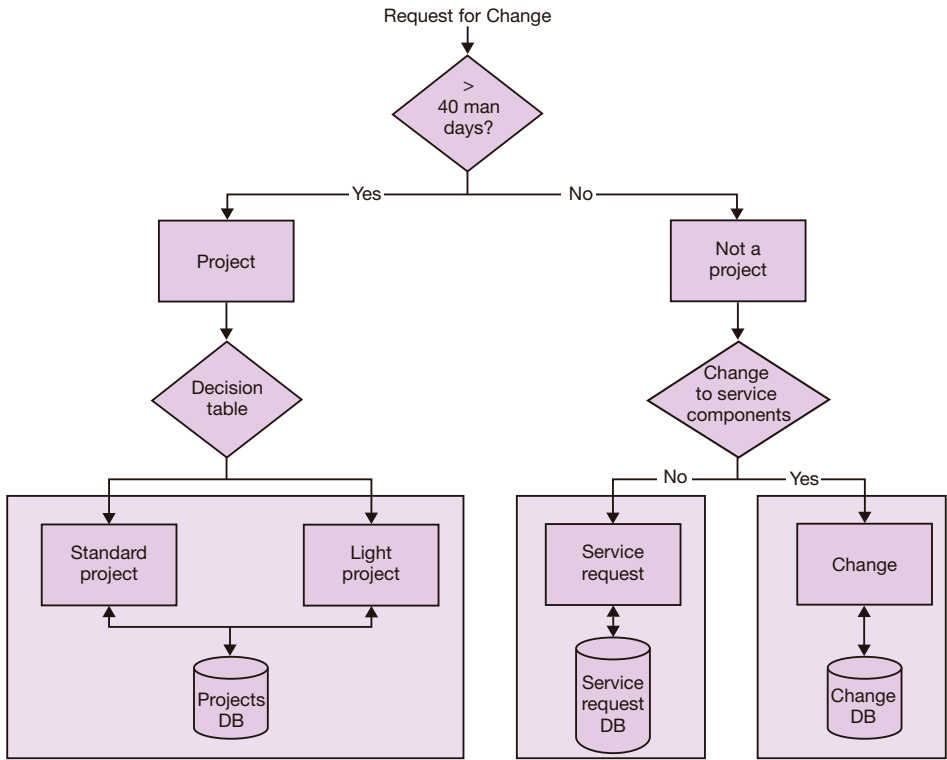


Figure 7 Translating requests into projects, changes or service requests

Making the choice between the “standard” or “light” model can be supported by a decision table. An example of such a table is shown in table 2, where the decision to opt for a model could be taken when two or more drivers are supporting it.

Driver	Light project	Standard project
Effort	40 < man days < 200	> 200 man days
Total cost	< 300k euro	> 300k euro
Strategic Business Plan initiative	No	Yes
Risk	Low to moderate (damage < 300k euro)	High (damage > 300k euro)
Others...		

Table 2 Drivers to choose a model for project management

When it is decided to manage a RFC as a project, the RFC should be closed accordingly and a project opened, filling the initial project mandate with RFC information. This is graphically illustrated in interface number 4 in figure 5.

The other interface we have to analyze in this section is the one from project to change management (number 5 in figure 6). This interface is probably the more complex and is

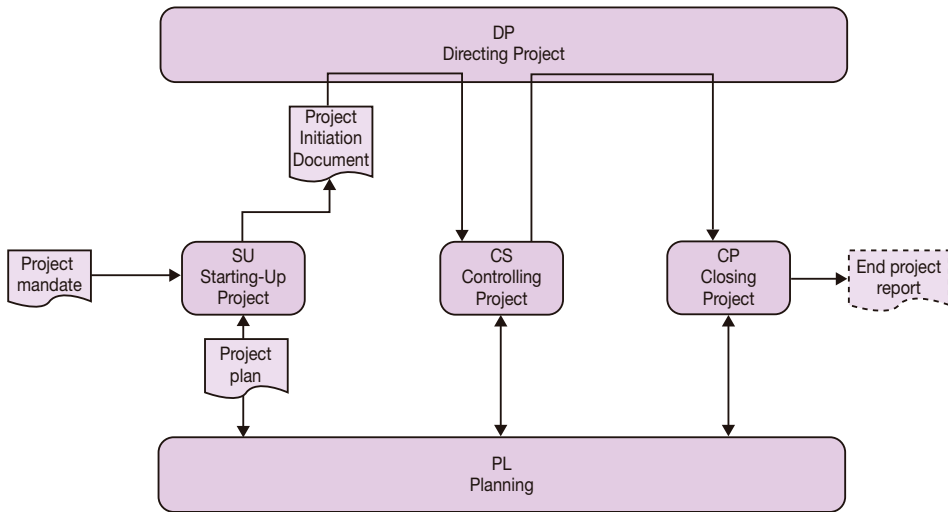


Figure 8 A simplified project management model based on PRINCE2 (Source: OGC)

activated when a project needs to modify the design of any existing service component, or when a new service is designed. When there is such a need, it is usual that certain change management activities are performed collectively in the project tasks. With reference to figure 1, it is common to see assessment, evaluation, planning, authorization, implementation and review tasks performed through project activities. When this happens and activities are executed by an ad hoc project organization not involving the change management functions, then there is a very high risk that there will be conflict between the project and change management objectives in a later phase of the project. This may happen because project management needs to pay attention to the project objectives, especially the timing and costs constraints, while change management, as part of service management, focuses upon the continued functioning of services in line with their requirements. A situation can arise, therefore, where project management are pushing to release a new component, whilst change management are resisting because of either late communication, incapacity to perform the necessary tests or inadequacy of the components to be released. This situation would be detected at the point when project management try to release deliverables to the production environment, as this task is normally under the control of change management.

In order to avoid this situation, there are some mandatory decisions to be taken and some alternatives to be chosen. The mandatory decision involves the change management function and early roles in the project. The alternatives derive from the relationships between the processes (project and change management).

The first alternative is to have an overall project plan where changes to IT services are identified and related activities are planned and updated according to the change management process. All information about these activities, such as timing and costs, should flow between the change and project domains. In this scenario, the change management process and function play an established role for the specific tasks which are part of the

changes, and the project activities are simply “embedding” and coordinating them. We can, therefore, say that project and change coexist: Project management determines changes which are executed by change management and taken into account by project management. The following is an example: the PRINCE2 “MP1 Accepting a Work Package” may be performed through the “Assess and Evaluate Change” and “Authorize Change” activities of change management process when a work package is related to one or more changes.

A second alternative is to manage the typical change management activities as normal project activities, using change management mainly for administrative aims if necessary (for example for homogeneous recording and tracking of changes). This means, for example, that changes may be collectively assessed by a project activity where the change management function is involved and where results are passed to change management in order to update change record information and status (if change management decides to formally open changes). The authorization may be managed as a project activity so that when changes come to the change authority for approval, this is only an information and administrative task, given that the records have been pre-approved by the project management. Implementation tasks (Work Orders generated in change management) are also managed as project tasks. Choosing this second scenario means that project management is given a more relevant role. Change management roles are involved in project management activities, but change management is not executed as an independent process, but rather it is updated based on project activities and results. An example is the following: different work packages (corresponding to changes) are requested by a “CS1 Authorising a Work Package” activity of the project. The change management function takes part to the “MP1 Accepting a Work Package” activity of the project where Work Packages are approved. Accordingly, as a result of these activities, the change management function will create and update the status of changes, if necessary, but without newly executing any assessment activity.

Given these two scenarios, the preferable one is the first which clearly recognizes the specific roles and scope of both processes. The problem arises when change management is not effective and/or efficient and is, therefore, unable to respond to the necessities and constraints of project management, or when the priorities of projects are not addressed adequately by change management. In such cases, the second alternative should be carefully evaluated and could emerge as more suitable.

From an organizational point of view, the responsibilities should be clear between the functions and roles that are managing project management and those managing change management, and communication channels should be well established. ITIL version 2 was not clear about organizational matters (only the main process roles, such as change manager, were identified, but no information was given on how to organize functions), but ITIL V3 has added some details as depicted in figure 9. In some IT organizations the Program and project management depicted in figure 9 is a staff function of the CIO, working for the Application development department as well as for Operations. The scenarios we have illustrated have detailed the “RFCs from projects, formal handover of CIs” relationship as depicted in gray in figure 9.

From a tool point of view, as we have previously remarked, the change management functionalities are normally part of the service management suite. Here it is not common to find features for cost management such as timesheet or expenses management, or sophisticated planning functions typical of project management tools. Having a full view of activities and the workload generated by change and project management is key information

for good decision making and effective planning. Therefore, if a single solution does not cover all all functions it will be important to implement some interfaces between the project and change management tools:

- alignment of common master data (such as resources, projects)
- alignment of forecast data regarding changes (activities with timing, resources, workload, status, details)
- alignment of actual efforts and costs

EXPLORING INTERFACES: RELEASE AND PROJECT MANAGEMENT

The first interface we are going to explore is the one between release and project management, depicted as number 3 in figure 5. This interface is, in some aspects, similar to the one between change and project management (number 4 in figure 5). A release, which is a collection of changes to be deployed, may be required by project management, or it can be generated by service management. The first case is out of scope for this specific interface, given that the release is already part of a project (we will explore this interface later in this section). Service management deals with releases periodically (for example for regular updates or bug fixing), packing changes into them, or for emergency releases. This latter case will not generate, by definition, any project. Depending on the type of changes packed into a release, the need to manage a specific release as a project may arise to further improve the likelihood of achieving the expected results. But in a well established service management organisation with properly set transition management functions (especially, with reference to figure 9, service knowledge management, service release and deployment, service test management) and having mature release management processes in place, this will probably not happen as these processes should be sufficient to guarantee a well managed transition of the release itself.

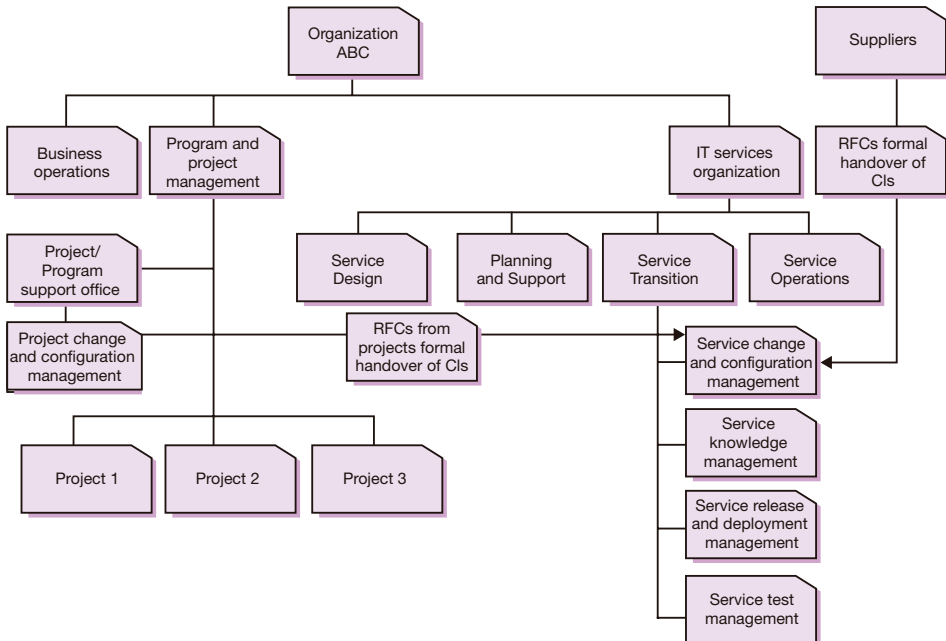


Figure 9 ITIL V3 example of Service Transition organization (Source: OGC)

In organizations where service management is not mature or adopted, at least as far as release management is concerned, then when a release is requested to be installed, service operations should evaluate if the usual practice and normal activities will be sufficient to deal safely with it. Again, we can rely on human experience and judgement to make such a decision, or we can define some thresholds. A useful one, if change management is in place, is the number of changes included in the release, by type (for example major, significant, minor). An example of the use of this type of driver as a threshold is given in table 3.

Driver	Threshold
Major changes	> 0
Significant changes	> 3
Minor changes	> 10

OR → project

Table 3 Example of use of number and types of changes as a threshold to manage releases as projects

If change management is not in place, the estimated effort (as with the interface between change and project) to complete all release management activities (planning, testing, deploying, and review) may be adopted and a threshold of forty man days may be reasonable.

Once it is agreed that a release has to be managed as a project, another decision may be necessary, i.e. the choice of project model. Again, it is possible to use a decision matrix similar to the one already described in table 2. Nevertheless, the probability that a full PRINCE2 approach is adopted is, in practice, very low, because such a high impact release would have probably been managed earlier as a project (hopefully, not when it is the time to deploy it!). So a light project approach, as described earlier (see figure 8) is probably preferable.

The next and final interface to explore is the one between project and release management, number 6 in figure 6. Projects, when dealing with IT applications and software, normally organize deployment in releases which implement some or the full set of requisites. If service management is in place (see interface 5 in figure 6), these requisites have already been mapped into changes and interface 6 practically corresponds to interface 1 (between change and release management). Project management knows from change management when the changes will be ready for testing and deployment, and requires release management to plan for these activities (this task is normally performed together by the project and release management organizations). All defined and supplied schedules will contribute to the project plan update.

Another possible scenario is similar to the one already described in alternative 2 for the project to change management interface. In this case, project management takes the lead in making the planning decisions, and change and release management are aligned to these project management decisions. The involvement of change and release management functions as early as possible is an important driver for the success of the project, because it ensures that all aspects of service transition will be considered and the necessary resources from change and release management functions will be involved as and when required.

In this scenario, not only the change activities but also the release activities are managed as part of the project. Planning, testing deployment, etc., are planned by the project

management organization and are executed according to project plans, which are defined in conjunction with change and release management staff. These staff will also ensure that the release management principles, such as the V-Model for testing, are applied to the project and will guarantee that other releases managed outside project boundaries will take place in compatible slots with the necessary resources.

As suggested for the interface between project and release management, the preferable scenario is the first one where project management interfaces to a mature service management organization, working closely with it to define the timing of activities according to its processes and letting it taking care of activities.

Finally, as already mentioned, project and release management tools are often distinct, the latter being embedded in software lifecycle management solutions. In practice it is often possible to manage the interfaces between the supported processes manually. Further improvements may arise through the implementation of an interface to automatically update project plans with release information (releases and their related activities may be loaded as project activities so that any change to the release schedule and allocated resources is immediately reflected in the project plan).

CONCLUSIONS

In this article we have clearly identified and explored the relationships between change, project and release management, assuming ITIL and PRINCE2 as reference frameworks to determine the scope and details of the processes. To this aim, a simple graphical model has been created.

The relationships between the service management processes (change and release) are traced in ITIL, especially in the latest version 3 of the framework, but not fully developed to be practically implemented. The article contains suggestions and practical examples useful to this aim.

Relationships between projects and the other processes are suggested as being necessary in the frameworks, but are not detailed even at a very rough level. They are more complex than those relationships between change and release management, primarily because they involve powerful but independent functions of the organization. Different implementation scenarios for the relationships are possible, depending on the maturity and influence of the corresponding functions, with project management being traditionally better established in application development units, whilst service management functions tend to be located in more of an operations context. The way the processes are interfaced can also be viewed as an indication of how the balance between stability versus responsiveness is established in the IT organization.

We have learned that changes should be managed as projects if their size and associated risks become relevant; less common is the need to manage releases as projects, as they can normally be addressed by release management processes or are already included in existing projects. Projects generate changes to implement specific project requirements and may use release management to test and deploy them.

The scenario where project management leverages service management processes and organizational units to develop requirements into changes, group them into releases and

have them implemented, tested and deployed is the preferable one for organizations where service management is mature; this optimizes the use of resources and the overall scheduling of activities, whilst minimizing the risks of adverse impact. But it is not the only possibility. Project management can also embed and substitute the typical activities of change and release management, applying the principles of the two disciplines and keeping them aligned with feed-back information.

Whatever solution is adopted, based on organizational structure and maturity, the secret of the success of the interfaces between project and change and project and release lies in good communication and involvement between the project management organizational units and the service management functions. An effective communication channel should be established as soon as possible between project management and project activities, and the timing should be compatible with change and release management principles and procedures.

In the practical implementation of interfaces, some issues must be addressed or they may threaten success:

- It is a pre-requisite to have mature service and project management processes and functions.
- The correct balance must be achieved between project management and service management responsibilities and spheres of influence.
- The interfaces must be designed in detail and based on real context, maturity and established processes and organization.
- All resources must be required to act according to the newly designed scenarios.
- Project and service management tools should be integrated to improve process efficiency and effectiveness.

Christian Cantù (Italy), economical engineer, is co-founder and partner at iCONS – Innovative Consulting, a specialist in IT service management and project management, and a member of ITSMF Italy.

Maxime Sottini (Italy), electronic engineer, is founder and Chief Executive Officer at iCONS – Innovative Consulting, a specialist in IT service management and project management, board member of itSMF Italy, and itSMF International IPESC and IQaC delegate for itSMF Italy and itSMF Italy responsible for Certifications and Qualifications.

REFERENCES

- Office of Government Commerce (2007). *ITIL: Service Transition*. London: The Stationary Office.
- Office of Government Commerce (2007). *ITIL: Service Support*. London: The Stationary Office.
- Office of Government Commerce (2007). *ITIL: Service Design*. London: The Stationary Office.
- Office of Government Commerce (2005). *ITIL: Managing Successful Projects with PRINCE2*. London: The Stationary Office.

Colophon

Title:	IT Service Management Global Best Practices – Volume 1
Editors:	Inform-IT, NL Jan van Bon (Chief Editor) Arjen de Jong Mike Pieper Ruby Tjassing Tieneke Verheijen Annelies van der Veen
Copy-editors:	Steve Newton, UK Jayne Wilkinson, UK
Editorial Board:	Dutch Society for Information Management: Rudolf Liefers EXIN International: Lex Hendriks Forrester Research: Peter O'Neill HP: Hans Bestebreurtje ISACA NL: Harry Boonen IT Skeptic (Rob England), New Zealand itSMF Australia: Karen Ferris itSMF Israel: Matiss Horodishtiano itSMF Italy: Maxime Sottini itSMF Japan: Takashi Yagi, supported by Reiko Morita itSMF South Africa: Peter Brooks National Health Services UK (NHS): Kevin Holland Norea NL: Ron Feijten Pink Elephant Canada: Troy DuMoulin Quint Wellington Redwood, now Siemens USA: Robert E. Matthews The Hague University of Professional Education: Marcel Spruit Tilburg University/Tias EDP-auditing & EDS: Jan Boogers Tot-Z NL: Ton van den Hoogen University of Antwerp Management School (UAMS): Steven De Haes
Publisher:	Van Haren Publishing (info@vanharen.net)
Edition:	First edition, second impression with small amendments, June 2008
ISBN:	Volume 1, 2008, 978 90 8753 100 3 Deel 4, 2007, ISBN 978 90 8753 043 3 Deel 3, 2006, ISBN 90 77212 74 4 Deel 2, 2005, ISBN 90 77212 44 2 Deel 1, 2004, ISBN 90 77212 17 5
Design & layout:	CO2 Premedia bv, Amersfoort – NL

© 2008, itSMF International

All rights reserved. No part of this publication may be reproduced in any form by print, photo print, microfilm or any other means without written permission by the publisher.

Although this publication has been composed with much care, neither author, nor editor, nor publisher can accept any liability for damage caused by possible errors and/or incompleteness in this publication.

TRADEMARK NOTICE

ITIL® is a Registered Trade Marks and Registered Community Trade Marks of the Office of Government Commerce, and is Registered in the U.S. Patent and Trademark Office.

Contact the editors for ideas, suggestions and improvements: Keith.Aldis@itsmf.co.uk

ITIL Books

The Official Books from itSMF



Foundations of IT Service Management Based on ITIL®V3

Now updated to encompass all of the implications of the V3 refresh of ITIL, the new V3 Foundations book looks at Best Practices, focusing on the Lifecycle approach, and covering the ITIL Service Lifecycle, processes and functions for Service Strategy, Service Design, Service Operation, Service Transition and Continual Service Improvement.

ISBN: 978 908753057 0 (ENGLISH EDITION)

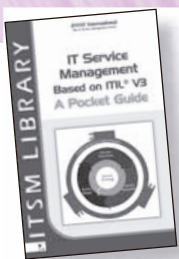
PRICE €39.95 EXCL TAX

Foundations of IT Service Management Based on ITIL®

The bestselling ITIL® V2 edition of this popular guide is available as usual, with 13 language options to give you the widest possible global perspective on this important subject.

ISBN: 978 907721258 5 (ENGLISH EDITION)

PRICE €39.95 EXCL TAX



IT Service Management Based on ITIL®V3: A Pocket Guide

A concise summary for ITIL®V3, providing a quick and portable reference tool to this leading set of best practices for IT Service Management.

ISBN: 978 908753102 7 (ENGLISH EDITION)

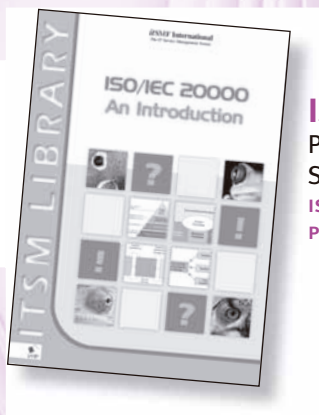
PRICE €14.95 EXCL TAX

Van Haren Publishing (VHP) is a leading international publisher, specializing in best practice titles for IT management and business management. VHP publishes in 14 languages, and has sales and distribution agents in over 40 countries worldwide: www.vanharen.net

Copyright protected. Use is for Single Users only via a VHP Approved License.
For information and printed versions please see www.vanharen.net

ISO/IEC 20000

The Official Books from itSMF



ISO/IEC 20000: An Introduction

Promoting awareness of the certification for organizations within the IT Service Management environment.

ISBN: 978 908753081 5 (ENGLISH EDITION)

PRICE €49.95 EXCL TAX

Implementing ISO/IEC 20000 Certification: The Roadmap

Practical advice, to assist readers through the requirements of the standard, the scoping, the project approach, the certification procedure and management of the certification.

ISBN: 978 908753082 2

PRICE €39.95 EXCL TAX



ISO/IEC 20000: A Pocket Guide

A quick and accessible guide to the fundamental requirements for corporate certification.

ISBN: 978 907721279 0 (ENGLISH EDITION)

PRICE €14.95 EXCL TAX

Other leading ITSM Books from itSMF



Metrics for IT Service Management

A general guide to the use of metrics as a mechanism to control and steer IT service organizations, with consideration of the design and implementation of metrics in service organizations using industry standard frameworks.

ISBN: 978 907721269 1

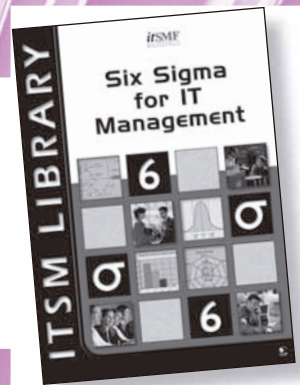
PRICE €39.95 EXCL TAX

Six Sigma for IT Management

The first book to provide a coherent view and guidance for using the Six Sigma approach successfully in IT Service Management, whilst aiming to merge both Six Sigma and ITIL® into a single unified approach to continuous improvement. Six Sigma for IT Management: A Pocket Guide is also available.

ISBN: 978 907721230 1 (ENGLISH EDITION)

PRICE €39.95 EXCL TAX

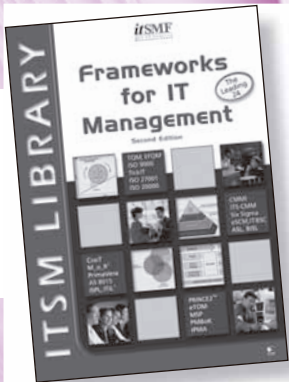


Frameworks for IT Management

An unparalleled guide to the myriad of IT management instruments currently available to IT and business managers. Frameworks for IT Management: A Pocket Guide is also available.

ISBN: 978 907721290 5 (ENGLISH EDITION)

PRICE €39.95 EXCL TAX

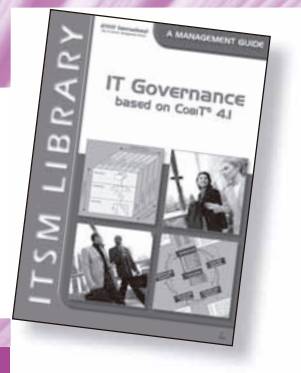


IT Governance based on CobIT 4.1: A Management Guide

Detailed information on the overall process model as well as the theory behind it.

ISBN: 978 90 8753116 4 (ENGLISH EDITION)

PRICE €20,75 EXCL TAX



Contact your local chapter for ITSM Library titles ...please see www.itsmfbooks.com for details.

Copyright protected. Use is for Single Users only via a VHP Approved License.
For information and printed versions please see www.vanharen.net