4 Service Transition processes

This chapter sets out the processes and activities on which effective Service Transition depends. These comprise both lifecycle processes and those almost wholly contained within Service Transition. Each is described in detail, setting out the key elements of that process or activity.

The processes and activities and their relationships are set out in Figure 2.3, and the topics specifically addressed in this chapter are:

- Transition Planning and Support
- Change Management
- Service Asset and Configuration Management
- Release and Deployment Management
- Service Validation and Testing
- Evaluation
- Knowledge Management.

Some of these processes are used throughout the service lifecycle, but are addressed in this publication since they are central to effective Service Transition.

The other processes and activities are mostly contained within the Service Transition phase of the lifecycle, but also are made use of in other phases, e.g. evaluation of design, and performance testing within operations.

The scope, goals, purpose and vision of Service Transition as a whole are set out in section 2.4.

4.1 TRANSITION PLANNING AND SUPPORT

4.1.1 Purpose, goals and objectives

The purpose of the Transition Planning and Support activities is to:

- Plan appropriate capacity and resources to package a release, build, release, test, deploy and establish the new or changed service into production
- Provide support for the Service Transition teams and people
- Plan the changes required in a manner that ensures the integrity of all identified customer assets, service assets and configurations can be maintained as they evolve through Service Transition
- Ensure that Service Transition issues, risks and deviations are reported to the appropriate stakeholders and decision makers
- Coordinate activities across projects, suppliers and service teams where required.

The goals of Transition Planning and Support are to:

- Plan and coordinate the resources to ensure that the requirements of Service Strategy encoded in Service Design are effectively realized in Service Operations
- Identify, manage and control the risks of failure and disruption across transition activities.

The objective of Transition Planning and Support is to:

- Plan and coordinate the resources to establish successfully a new or changed service into production within the predicted cost, quality and time estimates
- Ensure that all parties adopt the common framework of standard re-usable processes and supporting systems in order to improve the effectiveness and efficiency of the integrated planning and coordination activities
- Provide clear and comprehensive plans that enable the customer and business change projects to align their activities with the Service Transition plans.

4.1.2 Scope

The scope of the Service Transition Planning and Support activities includes:

- Incorporating design and operation requirements into the transition plans
- Managing and operating Transition Planning and Support activities
- Maintaining and integrating Service Transition plans across the customer, service and contract portfolios
- Managing Service Transition progress, changes, issues, risks and deviations
- Quality review of all Service Transition, release and deployment plans
- Managing and operating the transition processes, supporting systems and tools
- Communications with customers, users and stakeholders
- Monitoring and improving Service Transition performance.

4.1.3 Value to business

Effective Transition Planning and Support can significantly improve a service provider's ability to handle high volumes of change and releases across its customer base. An integrated approach to planning improves the alignment of the Service Transition plans with the customer, supplier and business change Project Plans.

4.1.4 Policies, principles and basic concepts

This section sets out basic concepts within that support for effective planning for Service Transition.

Service Design will – in collaboration with customers, external and internal suppliers and other relevant stakeholders – develop the Service Design and document it in a Service Design Package (SDP). The SDP includes the following information that is required by the Service Transition team:

- The applicable service packages (e.g. Core Service Package, Service Level Package)
- The service specifications
- The service models
- The architectural design required to deliver the new or changed Service including constraints
- The definition and design of each release package
- The detailed design of how the service components will be assembled and integrated into a release package
- Release and deployment plans
- The Service Acceptance Criteria.

4.1.4.1 Service Transition policy

Policies that support Service Transition are provided in Chapter 3.

The Change, Configuration and Knowledge Management policies also support Service Transition and further examples of these are provided in sections 4.2, 4.3 and 4.7.

4.1.4.2 Release policy

The release policy should be defined for one or more services and include:

- The unique identification, numbering and naming conventions for different types of release together with a description
- The roles and responsibilities at each stage in the release and deployment process
- The expected frequency for each type of release
- The approach for accepting and grouping changes into a release, e.g. how enhancements are prioritized for inclusion
- The mechanism to automate the build, installation and release distribution processes to improve re-use, repeatability and efficiency
- How the configuration baseline for the release is captured and verified against the actual release contents, e.g. hardware, software, documentation and knowledge
- Exit and entry criteria and authority for acceptance of the release into each Service Transition stage and into the controlled test, training, disaster recovery and production environments
- Criteria and authorization to exit early life support and handover to Service Operations.

A release that consists of many different types of service assets may involve many people, often from different organizations. The typical responsibilities for handover and acceptance of a release should be defined and then they can be modified as required for specific transitions. The main roles and responsibilities at points of handover should be defined to ensure that everyone understands their role and level of authority and those of others involved in the release and deployment process.

An example of a responsibility matrix for an organization that supports client–server applications is shown in Table 4.1. Such a matrix will help to identify gaps and overlaps and typical roles can be planned for the future.

Table 4.1 Example responsibility matrix for release points during Service Transition

	Development	Controlled test	Release to production	Production
Class of object	*Released from*	*Accepted by*	*Authority to release to live*	*Accepted and supported by*
Purchased package	Application development manager	Test manager	Change manager	Operations manager
Customized modules	Application development manager	Test manager	Change manager	Operations manager
Physical database changes	Application development manager	Database administrator	Change manager	Database administrator
Server	Server builder	Server manager	Change manager	Server manager
Desktop build (e.g. a new application)	Desktop development manager	Test manager	Change manager	Desktop support manager
Desktop application (already built and within operational constraints)	Desktop development manager	Desktop support manager	Desktop support manager, change manager	Desktop support manager
Desktop computers	Logistics	Desktop support	Desktop support manager, change manager	Desktop support manager
Desktop service	Service development	Desktop support	Service level management, desktop support manager, change manager	Service level management, desktop support manager
Release/Change authorization	Development manager	Test manager	Release manager, test manager, operations manager, desktop support service, desk user at each site, customer stakeholder, change manager	Service desk users

All releases should have a unique identifier that can be used by Configuration Management and the documentation standards. The types of release should be defined as this helps to set customer and stakeholder expectations about the planned releases. A typical example is:

- **Major releases**, normally containing large areas of new functionality, some of which may eliminate temporary fixes to problems. A major upgrade or release usually supersedes all preceding minor upgrades, releases and emergency fixes.
- **Minor releases**, normally containing small enhancements and fixes, some of which may already have been issued as emergency fixes. A minor upgrade or release usually supersedes all preceding emergency fixes.
- **Emergency releases**, normally containing the corrections to a small number of known errors or sometimes an enhancement to meet a high priority business requirement.

A release policy may say, for example, that only strict 'emergency fixes' will be issued in between formally planned releases of enhancements and non-urgent corrections.

An extract from a release policy is shown in Table 4.2, which shows how different types of release can be defined.

Table 4.2 Extract from a service release policy for a retail organization

SERVICE	Release definition*	Naming/Numbering	Frequency/Occurrence	Release window
Store service	Type A	SS_x	Annual (Feb)	Wednesday 01.00–04.00 hours
	Type B or C	SS_1.x or SS_1.1.x	Quarterly	Not holiday weekends
	Emergency	SS_1.1.1.x	As required	Not 1 September to 31 January
e-store web service	Type A	ESWnnn_x	6 months	01.00–02.00 hours
	Type B and C	ESWnnn_1.x	Monthly	Not holiday weekends
	Emergency	ESWnnn_1.1.x	As required	Not 1 October to 10 January
e-store delivery service	Type A	ESDnnn_x	6 months	01.00–02.00 hours
	Type B	ESDSnnn_1.x	Quarterly	Highest level of authorization required during holiday weekends
	Type C	ESDnnn_1.1.x	Monthly	
	Emergency	ESDnnn_1.1.1.x	As required	

*Release definitions

Type A	Something that impacts the whole system/service
Type B	A release that will impact part of the system, e.g. single sub-system or sub-service
Type C	Correction to a single function
Emergency	A change required to restore or continue service to ensure the Service Level Agreement (SLA) is maintained

4.1.5 Process activities, methods and techniques

4.1.5.1 Transition strategy

The organization should decide the most appropriate approach to Service Transition based on the size and nature of the core and supporting services, the number and frequency of releases required, and any special needs of the users – for example, if a phased roll-out is usually required over an extended period of time.

The Service Transition strategy defines the overall approach to organizing Service Transition and allocating resources. The aspects to consider are:

- Purpose, goals and objectives of Service Transition
- Context, e.g. service customer, contract portfolios
- Scope – inclusions and exclusions
- Applicable standards, agreements, legal, regulatory and contractual requirements:
    - Internal and externals standards
    - Interpretation of legislation, industry guidelines and other externally imposed requirements
    - Agreements and contracts that apply to Service Transition
- Organizations and stakeholders involved in transition:
    - Third parties, strategic partners, suppliers and service providers
    - Customers and users
    - Service Management
    - Service provider
    - Transition organization (see section 6.2)
- Framework for Service Transition:
    - Policies, processes and practices applicable to Service Transition including process service provider interfaces (SPIs)
    - Roles and responsibilities
    - Transition resource planning and estimation
    - Transition preparation and training requirements
    - The release and change authorization
    - Re-using the organization's experience, expertise, tools, knowledge and relevant historical data
    - Shared resources and service to support Service Transition
- Criteria:
    - Entry and exit criteria for each release stage
    - Criteria for stopping or re-starting transition activities

  - Success and failure criteria
- Identification of requirements and content of the new or changed service:
  - Services to be transitioned with target locations, customers and organizational units
  - Release definitions
  - Applicable SDP including architectural design
  - Requirements for environments to be used, locations, organizational and technical
  - Planning and management of environments, e.g. commissioning and decommissioning
- People:
  - Assigning roles and responsibilities including approvals
  - Assigning and scheduling training and knowledge transfer
- Approach:
  - Transition model including Service Transition lifecycle stages
  - Plans for managing changes, assets, configurations and knowledge
    - Baseline and evaluation points
    - Configuration audit and verification points
    - Points where RFCs should be raised
    - Use of change windows
  - Transition estimation, resource and cost planning
  - Preparation for Service Transition
  - Evaluation
  - Release packaging, build, deployment and early life support
  - Error handling, correction and control
  - Management and control – recording, progress monitoring and reporting
  - Service performance and measurement system
  - Key performance indicators and improvement targets
- Deliverables from transition activities including mandatory and optional documentation for each stage:
  - Transition plans
  - Change and Configuration Management Plan
  - Release policy, plans and documentation
  - Test plans and reports
  - Build plans and documentation
  - Evaluation plan and report
  - Deployment plans and reports
  - Transition closure report
- Schedule of milestones
- Financial requirements – budgets and funding.

Service Transition lifecycle stages

The SDP should define the lifecycle stages for a Service Transition, e.g.:

- Acquire and test input configuration items (CIs) and components
- Build and test
- Service release test
- Service operational readiness test
- Deployment
- Early life support
- Review and close service transition.

For each stage there will be exit and entry criteria and a list of mandatory deliverables from the stage.

4.1.5.2 Prepare for Service Transition

The Service Transition preparation activities include:

- Review and acceptance of inputs from the other service lifecycle stages
- Review and check the input deliverables, e.g. SDP, Service Acceptance Criteria and evaluation report (see paragraph 4.6.6)
- Identifying, raising and scheduling RFCs
- Checking that the configuration baselines are recorded in Configuration Management before the start of Service Transition (see paragraph 4.3.4.2)
- Checking transition readiness.

The configuration baselines help to fix a point in history that people can reference and apply changes to in a manner that is understandable. Any variance to the proposed service scope, Service Strategy requirements and Service Design baseline must be requested and managed through Change Management.

At a minimum, it should be accepted (by design, transition and stakeholders) that the Service Design and all the release units can be operated and supported within the predicted constraints and environment. The evaluation activity described in section 4.6 performs the evaluation of the SDP and Service Acceptance Criteria and provides a report to Change Management with recommendations on whether the RFC should be authorized.

4.1.5.3 Planning and coordinating Service Transition

Planning an individual Service Transition

The release and deployment activities should be planned in stages as details of the deployment might not be known in detail initially. Each Service Transition plan should be developed from a proven Service Transition model wherever possible. Although Service Design provides the initial plan, the planner will allocate specific resources to the activities and modify the plan to fit in with any new circumstances, e.g. a test specialist may have left the organization.

A Service Transition plan describes the tasks and activities required to release and deploy a release into the test environments and into production, including:

- Work environment and infrastructure for the Service Transition
- Schedule of milestones, handover and delivery dates
- Activities and tasks to be performed
- Staffing, resource requirements, budgets and time-scales at each stage
- Issues and risks to be managed
- Lead times and contingency.

Allocating resources to each activity and factoring in resource availability will enable the Service Transition planner to work out whether the transition can be deployed by the required date. If resources are not available, it may be necessary to review other transition commitments and consider changing priorities. Such changes need to be discussed with change and release management as this may affect other changes that may be dependents or prerequisites of the release.

Integrated planning

Good planning and management are essential to deploy a release across distributed environments and locations into production successfully. An integrated set of transition plans should be maintained that are linked to lower-level plans such as release, build and test plans. These plans should be integrated with the change schedule, release and deployment plans. Establishing good-quality plans at the outset enables Service Transition to manage and coordinate the Service Transition resources, e.g. resource allocation, utilization, budgeting and accounting.

An overarching Service Transition plan should include the milestone activities to acquire the release components, package the release, build, test, deploy, evaluate and proactively improve the service through early life support. It will also include the activities to build and maintain the services and IT infrastructure, systems and environments and the measurement system to support the transition activities.

Adopting programme and project management best practices

It is best practice to manage several releases and deployments as a programme, with each significant deployment run as a project. The actual deployment may be carried out by dedicated staff, as part of broader responsibilities such as operations or through a team brought together for the purpose. Elements of the deployment may be delivered through external suppliers, and suppliers may deliver the bulk of the deployment effort, for example in the implementation of an off-the-shelf system such as an ITSM support tool.

Significant deployments will be complex projects in their own right. The steps to consider in planning include the range of elements comprising that service, e.g. people, application, hardware, software, documentation and knowledge. This means that the deployment will contain sub-deployments for each type of element comprising the service.

Reviewing the plans

The planning role should quality review all Service Transition, release and deployment plans. Wherever possible, lead times should include an element of contingency and be based on experience rather than merely supplier assertion. This applies even more for internal suppliers where there is no formal contract. Lead times will typically vary seasonally and they should be factored into planning, especially for long time-frame transitions, where the lead times may vary between stages of a transition, or between different user locations.

Before starting the release or deployment, the Service Transition planning role should verify the plans and ask appropriate questions such as:

- Are these Service Transition and release plans up to date?
- Have the plans been agreed and authorized by all relevant parties, e.g. customers, users, operations and support staff?
- Do the plans include the release dates and deliverables and refer to related change requests, known errors and problems?
- Have the impacts on costs, organizational, technical and commercial aspects been considered?
- Have the risks to the overall services and operations capability been assessed?

- Has there been a compatibility check to ensure that the configuration items that are to be released are compatible with each other and with configuration items in the target environments?
- Have circumstances changed such that the approach needs amending?
- Were the rules and guidance on how to apply it relevant for current service and release packages?
- Do the people who need to use it understand and have the requisite skills to use it?
- Is the service release within the SDP and scope of what the transition model addresses?
- Has the Service Design altered significantly such that it is no longer appropriate?
- Have potential changes in business circumstances been identified? See example below.

Anticipating changed business circumstances

A new version of a retail organization's point of sale system was designed and ready for transition to the operational environment. Although the new version offers added features, most improvements related to ease of use, ease of support and maintainability of the software. The transition was originally scheduled for installation in September, but delays in third party suppliers meant the service fails ready for test and subsequent deployment in late November; due for installation two weeks after acceptance testing begins. The initially planned approach of involving 20% of user staff in acceptance trials and store disruption across the user base was no longer appropriate. With the Christmas sales boom imminent, such disruption was not appropriate, and would have been prevented by the annual change freeze. Instead, a longer, slower but less resource-intensive acceptance testing approach was selected with rollout to stores rescheduled for late January.

Where the transition approach does require rethinking and probable alteration, this should be delivered through the formal Change Management process, since the consideration of alternatives and agreement of the revised transition approach must be properly documented. However, for foreseeable scenarios, where the path of action is documented as an accepted reaction to the circumstances, authority to record and proceed with a change may be delegated to Service Transition or other appropriate party for approval, e.g. customer or project. For example, where the Service Transition milestone dates and release dates can be achieved with the same cost and resources with no impact on the service definition.

4.1.6 Provide transition process support

4.1.6.1 Advice

Service Transition should provide support for all stakeholders to understand and be able to follow the Service Transition framework of processes and supporting systems and tools. Although the planning and support team may not have the specialist resources to handle some aspects it is important that they can identify a relevant resource to help projects, e.g. specialists to set up Configuration Management or testing tools.

Projects should implement Service Transition activities and tasks in accordance with applicable Service Transition standards, policies and procedures. However, Project Managers are not always aware of the need to adopt these standards, policies and procedures. When new projects start up the Service Transition and planning and support role should proactively seek opportunities to establish the Service Transition processes into the project quickly – before alternative methods are adopted. Another approach is to work closely with the programme or Project Support and offer support to projects via this route.

4.1.6.2 Administration

The Service Transition Planning and Support role should provide administration for:

- Managing of Service Transition changes and work orders
- Managing issues, risks, deviations and waivers
- Managing support for tools and Service Transition processes
- Communications to stakeholders – e.g. logistics and deployment plans need to be communicated to all stakeholders
- Monitoring the Service Transition performance to provide input into Continual Service Improvement.

Changes that affect the agreed baseline configuration items are controlled through Change Management.

Plans and progress should be communicated and made available to relevant stakeholders. The stakeholder list is defined in the service package received from design and Service Transition should establish the continued relevance of that list, and update it as necessary.

4.1.6.3 Progress monitoring and reporting

Service Transition activities require monitoring against the intentions set out in the transition model and plan. Measuring and monitoring the release and deployment will (at the conclusion) establish if the transition is proceeding according to plan.

Maintaining an oversight of the actual transitions against the integrated Service Transition plans, release and change schedules is essential. It includes monitoring the progress of each transition periodically and at milestone or baseline points as well as receiving and chasing updates.

Management reports on the status of each transition will help to identify when there are significant variances from plan, e.g. for project management and the Service Management organization to make decisions and take action.

In many cases the transition plans will require amendment to bring them into line with a reality that has changed since design. This is not synonymous with bad design or error in selecting transition models, but merely a reflection of a dynamic environment.

4.1.7 Triggers, input and output, and inter-process interfaces

The trigger is an authorized RFC for a Service Transition. The inputs are:

- Authorized RFC
- Service Design package
- Release package definition and design specification
- Service Acceptance Criteria (SAC).

Outputs:

- Transition strategy
- Integrated set of Service Transition plans.

4.1.8 Key performance indicators and metrics

Primary key performance indicators (KPIs) for Transition Planning and Support include:

- The number of releases implemented that met the customer's agreed requirements in terms of cost, quality, scope, and release schedule (expressed as a percentage of all releases)
- Reduced variation of actual vs predicted scope, quality, cost and time
- Increased customer and user satisfaction with plans and communications that enable the business to align their activities with the Service Transition plans
- Reduction in number of issues, risks and delays caused by inadequate planning.

Other KPIs for an effective transition and support process include:

- Improved Service Transition success rate through improved scope and integration of the planning activities
- Better management information on the predicted vs actual performance and cost of Service Transition
- Improved efficiency and effectiveness of the processes and supporting systems, tools, knowledge, information and data to enable the transition of new and changed services, e.g. sharing tool licences
- Reduction in time and resource to develop and maintain integrated plans and coordination activities
- Project and service team satisfaction with the Service Transition practices.

4.2 CHANGE MANAGEMENT

Changes arise for a variety of reasons:

- Proactively, e.g. seeking business benefits such as reducing costs or improving services or increasing the ease and effectiveness of support
- Reactively as a means of resolving errors and adapting to changing circumstances.

Changes should be managed to:

- Optimize risk exposure (supporting the risk profile required by the business)
- Minimize the severity of any impact and disruption
- Be successful at the first attempt.

Such an approach will deliver direct benefit to the bottom line for the business by delivering early realization of benefits (or removal of risk), with a saving of money and time.

To make an appropriate response to all requests for change entails a considered approach to assessment of risk and business continuity, change impact, resource requirements, change authorization and especially to the realizable business benefit. This considered approach is essential to maintain the required balance between the need for change and the impact of the change.

This section provides information on the Change Management process and provides guidance that is scalable for:

- Different kinds and sizes of organizations
- Small and large changes required at each lifecycle stage
- Changes with major or minor impact

- Changes in a required timeframe
- Different levels of budget or funding available to deliver change.

4.2.1 Purpose, goals and objectives

The purpose of the Change Management process is to ensure that:

- Standardized methods and procedures are used for efficient and prompt handling of all changes
- All changes to service assets and configuration items are recorded in the Configuration Management System
- Overall business risk is optimized.

The goals of Change Management are to:

- Respond to the customer's changing business requirements while maximizing value and reducing incidents, disruption and re-work
- Respond to the business and IT requests for change that will align the services with the business needs.

The objective of the Change Management process is to ensure that changes are recorded and then evaluated, authorized, prioritized, planned, tested, implemented, documented and reviewed in a controlled manner.

4.2.2 Scope

Change can be defined in many ways. The definition of a service change is:

Service change

'The addition, modification or removal of authorized, planned or supported service or service component and its associated documentation.'

The scope of Change Management covers changes to baselined service assets and configuration items across the whole service lifecycle.

Each organization should define the changes that lie outside the scope of their service change process. Typically these might include:

- Changes with significantly wider impacts than service changes, e.g. departmental organization, policies and business operations – these changes would produce RFCs to generate consequential service changes
- Changes at an operational level such as repair to printers or other routine service components.

Figure 4.1 shows a typical scope for the service Change Management process for an IT department and how it interfaces with the business and suppliers at strategic, tactical and operational levels. It covers interfaces to internal and external service providers where there are shared assets and configuration items that need to be under Change Management. Service Change Management must interface with business Change Management (to the left in Figure 4.1), and with the supplier's Change Management (to the right in the figure). This may be an external supplier with a formal Change Management system, or with the project change mechanisms within an internal development project.

The Service Portfolio provides a clear definition of all current, planned and retired services. Understanding the Service Portfolio helps all parties involved in the Service Transition to understand the potential impact of the new or changed service on current services and other new or changed services.

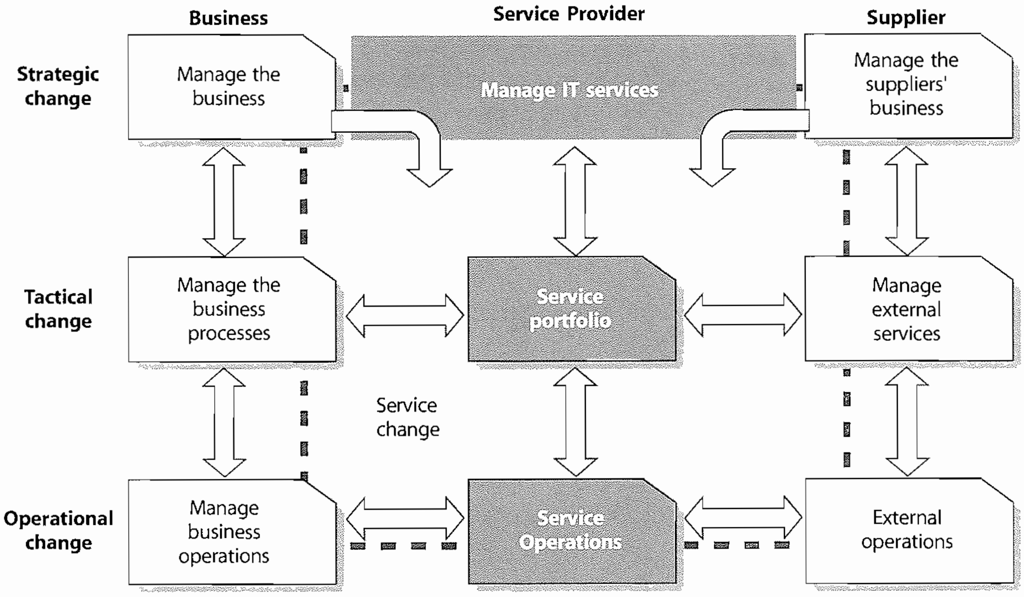


Figure 4.1 Scope of change and release management for services

Strategic changes are brought in via Service Strategy and the business relationship management processes. Changes to a service will be brought in via Service Design, Continual Service Improvement and the service level management process. Corrective change, resolving errors detected in services, will be initiated from Service Operations, and may route via support or external suppliers into a formal RFC.

Exclusions

This chapter does not cover strategic planning for business transformation or organizational change although the interfaces to these processes do need to be managed. Guidance on organizational change is addressed in Chapter 5. Business transformation is the subject of many publications aimed at the general business manager.

4.2.3 Value to business

Reliability and business continuity are essential for the success and survival of any organization. Service and infrastructure changes can have a negative impact on the business through service disruption and delay in identifying business requirements, but Change Management enables the service provider to add value to the business by:

- Prioritizing and responding to business and customer change proposals
- Implementing changes that meet the customers' agreed service requirements while optimizing costs
- Contributing to meet governance, legal, contractual and regulatory requirements
- Reducing failed changes and therefore service disruption, defects and re-work
- Delivering change promptly to meet business timescales
- Tracking changes through the service lifecycle and to the assets of its customers
- Contributing to better estimations of the quality, time and cost of change
- Assessing the risks associated with the transition of services (introduction or disposal)
- Aiding productivity of staff through minimizing disruptions due to high levels of unplanned or 'emergency' change and hence maximising service availability
- Reducing the Mean Time to Restore Service (MTRS), via quicker and more successful implementations of corrective changes
- Liaising with the business change process to identify opportunities for business improvement.

Example of IT service initiated business change

In the retail industry, bar-coding of goods coupled with bar-code readers at the check-out was initially introduced to deliver savings by removing the need to label every item, automating stock control, speeding customer throughput and reducing check-out staff. Suggestions from IT to the business resulted in making use of this facility to power innovative concepts such as Buy One Get One Free and capturing data on each individual's purchasing habits.

The reliance on IT Services and underlying information technology is now so complex that considerable time can be spent on:

- Assessing the impact of business change on IT
- Analysing the impact of a service or IT change on the business
- Notifying affected parties (of what is proposed, planned and implemented)
- Recording and maintaining accurate change, configuration, release and deployment records
- Managing and resolving incidents caused by change
- Identifying the problems that continually arise that require more change
- Introducing the new ideas and technology that cause even more change.

There are therefore considerable cost saving and efficiencies to be gained from well-structured and planned changes and releases.

As there is so much focus today on enterprise risk management, Change Management is a key process that comes under the scrutiny of auditors. The Institute of Internal Auditors, *Global Technology Audit Guide, Change and Patch Management Controls: Critical for Organizational Success*, provides guidance on assessing Change Management capability of an organization. It identifies risk indicators of poor Change Management that apply to business and IT change:

> 'By managing changes, you manage much of the potential risk that changes can introduce'
>
> The top five risk indicators of poor Change Management are:
>
> - Unauthorized changes (above zero is unacceptable)
> - Unplanned outages
> - A low change success rate
> - A high number of emergency changes
> - Delayed project implementations.

The following paragraph is extracted from the guide:

> What do all high-performing IT organizations have in common? They have a culture of Change Management that prevents and deters unauthorized change. They also 'trust but verify' by using independent detective controls to reconcile production changes with authorized changes, and by ruling out change first in the repair cycle during outages. Finally, they also have the lowest mean time to repair (MTTR). Auditors will appreciate that in these high-performing IT organizations, Change Management is not viewed as bureaucratic, but is instead the only safety net preventing them from becoming a low-performer In other words, IT management owns the controls to achieve its own business objectives, efficiently and effectively.
>
> Achieving a change success rate over 70 percent is possible only with preventive and detective controls.

Note: Mean Time to Restore Service (MTRS) should be used to avoid the ambiguity of Mean Time To Repair (MTTR). Although MTTR is a widely accepted industry term, in some definitions 'repair' includes only repair time but in others includes recovery time. The downtime in MTRS covers all the contributory factors that make the service, component or CI unavailable. MTRS is a measure of how quickly and effectively a service, component or CI can be restored to normal working after a failure and should be calculated using the following formula:

$$\text{MTRS (hours)} = \frac{\text{Total Downtime (hours)}}{\text{Number of service breaks}}$$

4.2.4 Policies, principles and basic concepts

This section sets out basic concepts within Change Management that support its effective execution.

4.2.4.1 Policies

Increasing the success rate of changes and releases requires Executive support for implementing a culture that sets stakeholder expectations about changes and releases and reduces unplanned work.

Pressure will be applied to reduce timescales and meet deadlines; to cut budgets and running costs; and to compromise testing. This must not be done without due diligence to governance and risk. The Service Transition management team will be called on from time to time to make a 'no go' decision and not implement a required change. There must be policies and standards defined which make it clear to the internal and external providers what must be done and what the consequence of non-adherence to policy will be.

Policies that support Change Management include:

- Creating a culture of Change Management across the organization where there is zero tolerance for unauthorized change
- Aligning the service Change Management process with business, project and stakeholder Change Management processes
- Prioritization of change, e.g. innovation vs preventive vs detective vs corrective change
- Establishing accountability and responsibilities for changes through the service lifecycle
- Segregation of duty controls
- Establishing a single focal point for changes in order to minimize the probability of conflicting changes and potential disruption to the production environment
- Preventing people who are not authorized to make a change from having access to the production environment
- Integration with other Service Management processes to establish traceability of change, detect unauthorized change and identify change related incidents
- Change windows – enforcement and authorisation for exceptions
- Performance and risk evaluation of all changes that impact service capability
- Performance measures for the process, e.g. efficiency and effectiveness.

4.2.4.2 Design and planning considerations

The Change Management process should be planned in conjunction with Release and Configuration Management. This helps the service provider to evaluate the impact of the change on the current and planned services and releases.

The requirements and design for the Change Management processes include:

- Requirements, e.g. to comply with relevant legislation, industry codes of practice, standards and organizational practices
- Approach to eliminating unauthorized change
- Identification and classification:
  - Change document identifiers
  - Change document types, change documentation templates and expected content
  - Impact, urgency, priorities

- Organization, roles and responsibilities:
  - Accountabilities and responsibilities of all stakeholders
  - Approach to independent testing and evaluation of change
  - Change authorization – levels of authorization and rules that govern decision making and actions, e.g. escalation
  - Composition of Advisory Boards, e.g. the Change Advisory Board (CAB) and the Emergency CAB (ECAB)
- Stakeholders:
  - Planning of changes and releases to enable stakeholders to make their own preparation and plan their activities
  - Communicating changes, change schedule and release plans
- Grouping and relating changes:
  - Into a release, build or baseline
  - By linking several child RFCs to a master RFC
- Procedures:
  - Change authorization policies, rules and procedures
  - For raising an RFC, including preparation and submission of change proposal
  - How change requests are tracked and managed, i.e. change records
  - How change requests are impact assessed and evaluated promptly
  - Identifying dependencies and incompatibilities between changes
  - For verifying the implementation of a change
  - Overseeing and evaluating deliverables from change and release implementation
  - To review changes regularly to identify trends and improvements, e.g. in the success or failure of changes and releases
- Interfaces to other Service Management processes, e.g. service level management and capacity management for impact assessment and review
- Approach to interfacing Change, Release and Configuration Management with the problem and incident management processes to measure and reduce change-related incidents.
- Configuration Management interfaces:
  - To provide quality information for impact assessment and reporting, e.g. comparison of As-Is to As-Planned configuration
  - To identify high-risk, high-impact CIs
  - To capture CIs, configuration baselines and releases
  - To capture related deliverables, e.g. Acceptance Criteria, test and evaluation reports.

4.2.4.3 Types of change request

A change request is a formal communication seeking an alteration to one or more configuration items. This could take several forms, e.g. 'Request for Change' document, service desk call, Project Initiation Document. Different types of change may require different types of change request. An organization needs to ensure that appropriate procedures and forms are available to cover the anticipated requests. Avoiding a bureaucratic approach to documenting a minor change removes some of the cultural barriers to adopting the Change Management process.

As much use as possible should be made of devolved authorization, both through the standard change procedure (see paragraph 4.2.4.4) and through the authorization of minor changes by Change Management staff.

During the planning of different types of change requests, each must be defined with a unique naming convention, (see paragraph 4.3.5.3). Table 4.3 provides examples of different types of change request across the service lifecycle.

For different change types there are often specific procedures, e.g. for impact assessment and change authorization.

4.2.4.4 Change process models and workflows

Organizations will find it helpful to predefine change process models – and apply them to appropriate changes when they occur. A process model is a way of predefining the steps that should be taken to handle a process (in this case a process for dealing with a particular type of change) in an agreed way. Support tools can then be used to manage the required process. This will ensure that such changes are handled in a predefined path and to predefined timescales.

Changes that require specialized handling could be treated in this way, such as emergency changes that may have different authorization and may be documented retrospectively.

Table 4.3 Example of types of request by service lifecycle stage

Type of change with examples	Documented work procedures	Service Strategy	Service Design	Service Transition	Service Operation	Continual Service Improvement
Request for Change to Service Portfolios – New portfolio line item – To predicted scope, Business Case, baseline – Service pipeline	Service Change Management	✓				
Request for Change to Service or service definition – To existing or planned service attributes – Project change that impacts Service Design, e.g. forecasted warranties – Service improvement	Service Change Management	✓	✓	✓	✓	✓
Project change proposal – Business change – No impact on service or design baseline	Project Change Management procedure		✓	✓		✓
User access request	User access procedure				✓	
Operational activity – Tuning (within specification/constraints) – Re-boot hardware on failure if no impact on other services – Planned maintenance	Local procedure (often pre-authorized – see paragraph 4.2.4.4)				✓	

The change process model includes:

- The steps that should be taken to handle the change including handling issues and unexpected events
- The chronological order these steps should be taken in, with any dependences or co-processing defined
- Responsibilities; who should do what
- Timescales and thresholds for completion of the actions
- Escalation procedures; who should be contacted and when.

These models are usually input to the Change Management support tools in use and the tools then automate the handling, management, reporting and escalation of the process.

4.2.4.5 Standard changes (pre-authorized)

A standard change is a change to a service or infrastructure for which the approach is pre-authorized by Change Management that has an accepted and established procedure to provide a specific change requirement.

Examples might include an upgrade of a PC in order to make use of specific standard and pre-budgeted software, new starters within an organization, or a desktop move for a single user. Other examples include low impact, routine application change to handle seasonal variation.

Approval of each occurrence of a standard change will be granted by the delegated authority for that standard change, e.g. by the budget holding customer for installation of software from an approved list on a PC registered to their organizational unit or by the third party engineer for replacement of a faulty desktop printer.

The crucial elements of a standard change are that:

- There is a defined trigger to initiate the RFC
- The tasks are well known, documented and proven
- Authority is effectively given in advance
- Budgetary approval will typically be preordained or within the control of the change requester
- The risk is usually low and always well understood.

Once the approach to manage standard changes has been agreed, standard change processes and associated change workflows should be developed and communicated. A change model would normally be associated with each standard change to ensure consistency of approach.

Standard changes should be identified early on when building the Change Management process to promote efficiency. Otherwise, a Change Management implementation can create unnecessarily high levels of administration and resistance to the Change Management process.

All changes, including standard changes, will have details of the change recorded. For some standard changes this may be different in nature from normal change records.

Some standard changes to configuration items may be tracked on the asset or configuration item lifecycle, particularly where there is a comprehensive CMS that provides reports of changes, their current status, the related configuration items and the status of the related CI versions. In these cases the Change and Configuration Management reporting is integrated and Change Management can have 'oversight' of all changes to service CIs and release CIs.

Some standard changes will be triggered by the request fulfilment process and be directly recorded and passed for action by the service desk.

4.2.5 Remediation planning

No change should be approved without having explicitly addressed the question of what to do if it is not successful. Ideally, there will be a back-out plan, which will restore the organization to its initial situation, often through the reloading of a baselined set of CIs, especially software and data. However, not all changes are reversible, in which case an alternative approach to remediation is required. This remediation may require a revisiting of the change itself in the event of failure, or may be so severe that it requires invoking the organization's business continuity plan. Only by considering what remediation options are available before instigating a change, and by establishing that the remediation is viable (e.g. it is successful when tested), can the risk of the proposed change be determined and the appropriate decisions taken.

4.2.6 Process activities, methods and techniques

This section provides approaches to managing service changes effectively by addressing the tasks carried out to achieve and deliver controlled change.

Overall Change Management activities include:

- Planning and controlling changes
- Change and release scheduling
- Communications
- Change decision making and change authorization
- Ensuring there are remediation plans
- Measurement and control

- Management reporting
- Understanding the impact of change
- Continual improvement.

Typical activities in managing individual changes are:

- Create and record the RFC
- Review RFC and change proposal:
  - Filter changes (e.g. incomplete or wrongly routed changes)
- Assess and evaluate the change:
  - Establish the appropriate level of change authority
  - Establish relevant areas of interest (i.e. who should be involved in the CAB)
  - Assess and evaluate the business justification, impact, cost, benefits and risk of changes
  - Request independent evaluation of a change (see 4.2.6.4)
- Authorize the change:
  - Obtain authorization/rejection
  - Communicate the decision with all stakeholders, in particular the initiator of the Request for Change
- Plan updates
- Coordinate change implementation
- Review and close change:
  - Collate the change documentation, e.g. baselines and evaluation reports
  - Review the change(s) and change documentation
  - Close the change document when all actions are completed.

Throughout all the process activities listed above and described within this section, information is gathered, recorded in the CMS and reported.

Figure 4.2 shows an example of a change to the service provider's services, applications or infrastructure. Examples of the states of the RFC are shown in italics. Change and configuration information is updated all the way through the activities. Figures 4.3 and 4.4 show the equivalent process flow for some examples of standard change process flows.

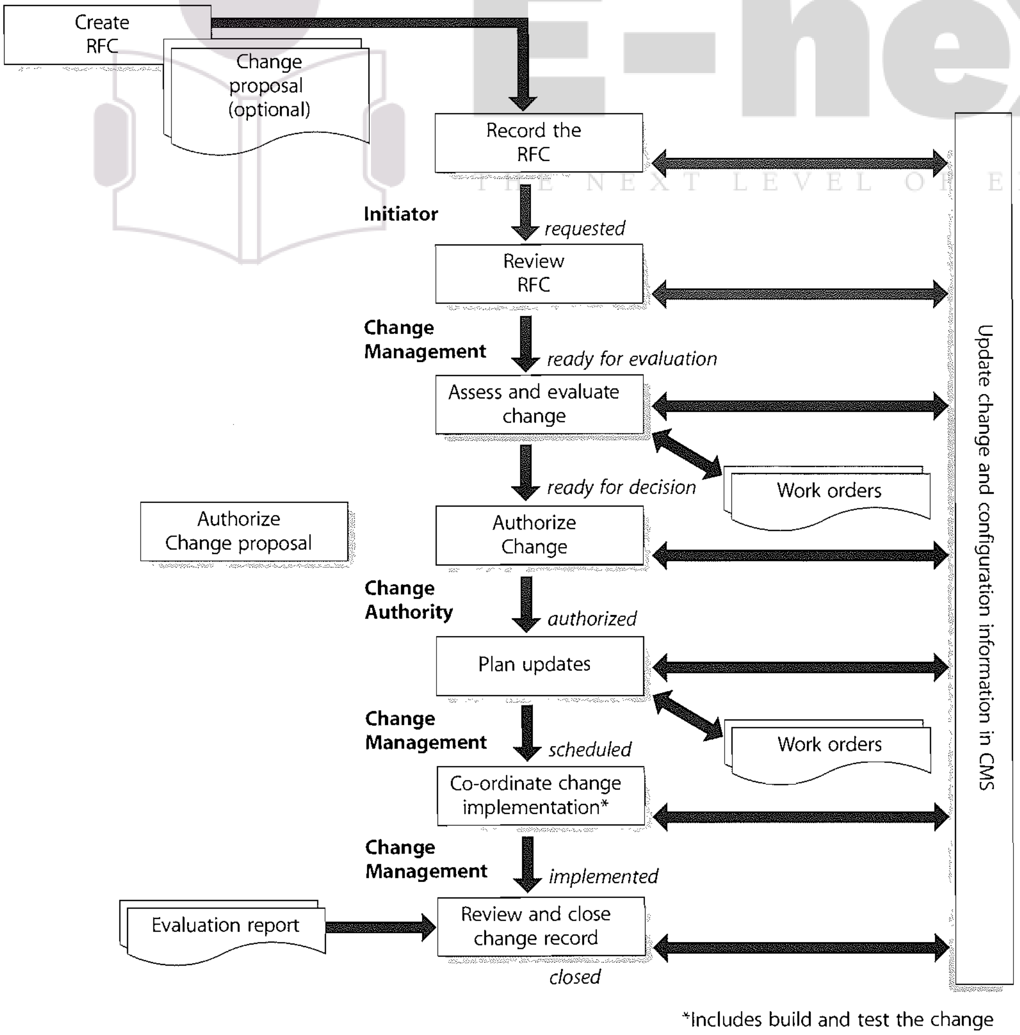


Figure 4.2 Example process flow for a normal change

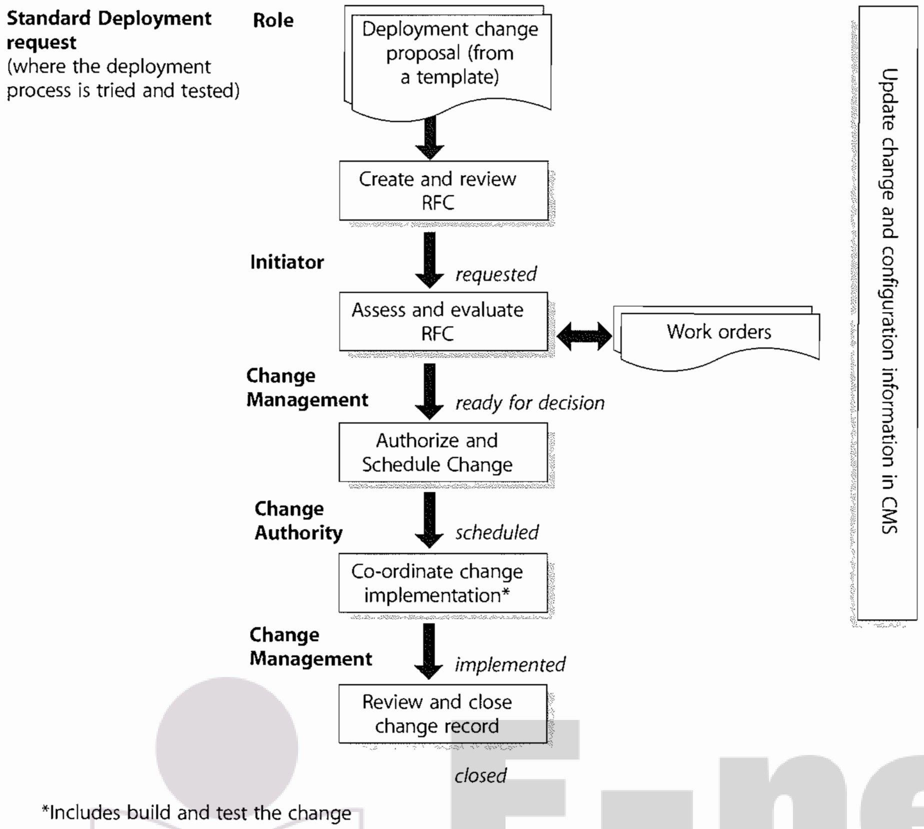

Standard Deployment request
(where the deployment process is tried and tested)
Role
Deployment change proposal (from a template)
Create and review RFC
Initiator
requested
Assess and evaluate RFC
Work orders
Change Management
ready for decision
Authorize and Schedule Change
Change Authority
scheduled
Co-ordinate change implementation*
Change Management
implemented
Review and close change record
closed
*Includes build and test the change
Update change and configuration information in CMS

Figure 4.3 Example process flow for standard deployment request

4.2.6.1 Normal Change Procedure

The text in this section sets out in detail the aspects followed within a Normal Change. The general principles set out apply to all changes, but where normal change procedure can be modified, i.e. for standard or emergency changes, this is set out following the explanation of normal change procedure.

4.2.6.2 Create and record Requests for change

The change is raised by a request from the initiator – the individual or organizational group that requires the change. For example, this may be a business unit that requires additional facilities, or problem management staff instigating an error resolution from many other sources.

For a major change with significant organizational and/or financial implications, a change proposal may be required, which will contain a full description of the change together with a business and financial justification for the proposed change. The change proposal will include sign-off by appropriate levels of business management.

Table 4.4 shows an example of the information recorded for a change; the level of detail depends on the size and impact of the change. Some information is recorded when the document is initiated and some information may be updated as the change document progresses through its lifecycle. Some information is recorded directly on the

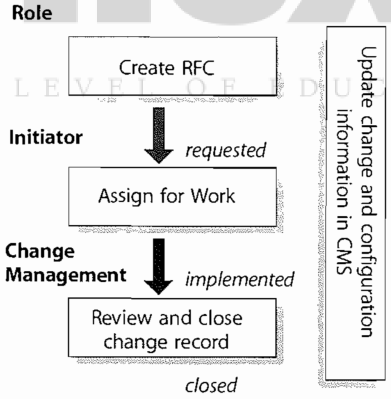


Figure 4.4 Example process flow for standard operational change request

Request for Change form and details of the change and actions may be recorded in other documents and referenced from the RFC, e.g. Business Case, impact assessment report.

Table 4.4 Example of contents of change documentation

Attribute on the change record	RFC	Change proposal (if appropriate)	Related assets/CIs
Unique number	✓		
Trigger (e.g. to purchase order, problem report number, error records, business need, legislation)	✓		
Description	Summary	Full description	
Identity of item(s) to be changed – description of the desired change	Summary	Full description	Service (for enhancement) or CI with errors (corrective changes)
Reason for change, e.g. Business Case	Summary	Full justification	
Effect of not implementing the change (business, technical, financial etc.)	✓		
Configuration items and baseline versions to be changed	✓	Affected baseline/release	Details of CIs in baseline/release
Contact and details of person proposing the change	✓		
Date and time that the change was proposed	✓		
Change category, e.g. minor, significant, major	Proposed		
Predicted timeframe, resources, costs and quality of service	Summary/reference	Full	
Change priority	Proposed		
Risk assessment and risk management plan	Summary/reference	Full	
Back-out or remediation plan	Possibly	Full	
Impact assessment and evaluation – resources and capacity, cost, benefits	Provisional	Initial impact	✓
Would the change require consequential amendment of IT Service Continuity Management (ITSCM) plan, capacity plan, security plan, test plan?	✓		Plans affected
Change decision body	✓		

Table 4.4 Example of contents of change documentation (continued)

Attribute on the change record	RFC	Change proposal (if appropriate)	Related assets/CIs
Decision and recommendations accompanying the decision	✓		
Authorization signature (could be electronic)	✓		
Authorization date and time	✓		
Target baseline or release to incorporate change into	✓		
Target change plan(s) for change to be incorporated into	✓		
Scheduled implementation time (change window, release window or date and time)	✓		
Location/reference to release/implementation plan	✓		
Details of change implementer	✓		
Change implementation details (success/fail/remediation)	✓		✓
Actual implementation date and time	✓		
Review date(s)	✓		
Review results (including cross-reference to new RFC where necessary)	Summary		
Closure	Summary		

The change record holds the full history of the change, incorporating information from the RFC and subsequently recording agreed parameters such as priority and authorization, implementation and review information. There may be many different types of change records used to record different types of change. The documentation should be defined during the process design and planning stage.

Different types of change document will have different sets of attributes to be updated through the lifecycle. This may depend on various factors such as the change process model and change category but it is recommended that the attributes are standardized wherever possible to aid reporting.

Some systems use work orders to progress the change as this enables complete traceability of the change. For example work orders may be issued to individuals or teams to do an impact assessment or to complete work required for a change that is scheduled for a specific time or where the work is to be done quickly.

As an RFC proceeds through its lifecycle, the change document, related records (such as work orders) and related configuration items are updated in the CMS, so that there is visibility of its status. Estimates and actual resources, costs and outcome (success or failure) are recorded to enable management reporting.

Change logging

The procedures for logging and documenting RFCs should be decided. RFCs might be able to be submitted on paper forms, through e-mail or using a web-based interface, for example. Where a computer-based support tool is used, the tool may restrict the format.

All RFCs received should be logged and allocated an identification number (in chronological sequence). Where change requests are submitted in response to a trigger such as a resolution to a problem record (PR), it is important that the reference number of the triggering document is retained to provide traceability.

It is recommended that the logging of RFCs is done by means of an integrated Service Management tool, capable of storing both the data on all assets and CIs and also, importantly, the relationships between them. This will greatly assist when assessing the likely impact of a change to one component of the system on all other components. All actions should be recorded, as they are carried out, within the Change Management log. If this is not possible for any reason, then they should be manually recorded for inclusion at the next possible opportunity.

Procedures will specify the levels of access and who has access to the logging system. While any authorized personnel may create, or add reports of progress to, an RFC (though the support tool should keep Change Management aware of such actions) only Change Management staff will have permission to close an RFC.

4.2.6.3 Review the Request for Change

The procedures should stipulate that, as changes are logged, Change Management should briefly consider each request and filter out any that seem to be:

- Totally impractical
- Repeats of earlier RFCs, accepted, rejected or still under consideration
- Incomplete submissions, e.g. inadequate description, without necessary budgetary approval.

These should be returned to the initiator, together with brief details of the reason for the rejection, and the log should record this fact. A right of appeal against rejection should exist, via normal management channels, and should be incorporated within the procedures.

4.2.6.4 Assess and evaluate the change

The potential impact on the services of failed changes and their impact on service assets and configurations need to be considered. Generic questions (e.g. the 'seven Rs') provide a good starting point.

The seven Rs of Change Management

The following questions must be answered for all changes. Without this information, the impact assessment cannot be completed, and the balance of risk and benefit to the live service will not be understood. This could result in the change not delivering all the possible or expected business benefits or even of it having a detrimental, unexpected effect on the live service.

- Who RAISED the change?
- What is the REASON for the change?
- What is the RETURN required from the change?
- What are the RISKS involved in the change?
- What RESOURCES are required to deliver the change?
- Who is RESPONSIBLE for the build, test and implementation of the change?
- What is the RELATIONSHIP between this change and other changes?

Many organizations develop specific impact assessment forms to prompt the impact assessors about specific types of change. This can help with the learning process, particularly for new services or when implementing a formal impact assessment step for the first time.

Responsibility for evaluating major change should be defined. It is not a best-practice issue because organizations are so diverse in size, structure and complexity that there is not a universal solution appropriate to all organizations. It is, however, recommended that major change is discussed at the outset with all stakeholders in order to arrive at sensible boundaries of responsibility and to improve communications.

Although Change Management is responsible for ensuring that changes are assessed and, if authorized, subsequently developed, tested, implemented and reviewed, clearly final responsibility for the IT service – including changes to it – will rest with the service manager and the service owner. They control the funding available and will have been involved in the change process through direct or delegated membership of the CAB.

When conducting the impact and resource assessment for RFCs referred to them, Change Management, CAB, ECAB or any others (nominated by Change Management or CAB members) who are involved in this process should consider relevant items, including:

- the impact that the change will make on the customer's business operation
- the effect on the infrastructure and customer service, as defined in the service requirements baselines, service model, SLA, and on the capacity and performance, reliability and resilience, contingency plans, and security
- the impact on other services that run on the same infrastructure (or on projects)
- the impact on non-IT infrastructures within the organization – for example, security, office services, transport, customer help desks
- the effect of not implementing the change
- the IT, business and other resources required to implement the change, covering the likely costs, the number and availability of people required, the elapsed time, and any new infrastructure elements required
- the current change schedule (CS) and projected service outage (PSO)
- additional ongoing resources required if the change is implemented
- impact on the continuity plan, capacity plan, security plan, regression test scripts and data and test environment, Service Operations practices.

No change is without risk

Simple changes may seem innocuous but can cause damage out of all apparent proportion to their complexity. There have been several examples in recent years of high profile and expensive business impact caused by the inclusion, exclusion or misplacing of a '.' in software code.

It is best practice to use a risk-based assessment during the impact assessment of a change or set of changes. For example the risk for:

- An individual change
- A set of changes implemented together
- Impacting the timescales of authorized changes on change and release schedules.

Table 4.5 Example of a change impact and risk categorization matrix

	Change impact/risk categorization matrix	
Change impact	High impact Low probability Risk category: 2	High impact High probability Risk category: 1
	Low impact Low probability Risk category: 4	Low impact High probability Risk category: 3
	Probability	

The focus should be on identifying the factors that may disrupt the business, impede the delivery of service warranties or impact corporate objectives and policies. The same disciplines used for corporate risk management or in project management can be adopted and adapted.

Risk categorization

The issue of risk to the business of any change must be considered prior to the authorisation of any change. Many organizations use a simple matrix like the one shown in Table 4.5 to categorize risk, and from this the level of change assessment and authorization required.

The relevant risk is the risk to the business service and changes require thorough assessment, wide communication, and appropriate authorization by the person or persons accountable for that business service. Assessing risk from the business perspective can produce a correct course of action very different from that which would have been chosen from an IT perspective, especially within high-risk industries.

High-risk industry

In one volatile and competitive business environment, the mobile telephone supply business, customers asked IT if they were now able to implement a much-needed change to the business software. The reply was that it could not go forward to the next change window because there was still a 30% risk of failure. Business reaction was to insist on implementation, for in their eyes a 70% chance of success, and the concomitant business advantage, was without any hesitation the right and smart move. Very few of their business initiatives had that high a chance of success.

The point is that the risk and gamble of the business environment (selling mobile telephones) had not been understood within IT, and inappropriate (i.e. IT) rules had been applied.

The dominant risk is the business one and that should have been sought, established, understood and applied by the service provider. Sensibly, of course, this might well be accompanied by documentation of the risk-based decision but nonetheless the need remains to understand the business perspective and act accordingly.

Evaluation of change

Based on the impact and risk assessments, and the potential benefits of the change, each of the assessors should evaluate the information and indicate whether they support approval of the change. All members of the change authority should evaluate the change based on impact, urgency, risk, benefits and costs. Each will indicate whether they support approval and be prepared to argue their case for any alterations that they see as necessary.

Allocation of priorities

Prioritization is used to establish the order in which changes put forward should be considered.

Every RFC will include the originator's assessment of the impact and urgency of the change.

The priority of a change is derived from the agreed impact and urgency. Initial impact and urgency will be suggested by the change initiator but may well be modified in the change authorization process. Risk assessment is of crucial importance at this stage. The CAB will need information on business consequences in order to assess effectively the risk of implementing or rejecting the change.

Impact is based on the beneficial change to the business that will follow from a successful implementation of the change, or on the degree of damage and cost to the business due to the error that the change will correct. The impact may not be expressed in absolute terms but may depend on the probability of an event or circumstance; for example a service may be acceptable at normal throughput levels, but may deteriorate at high usage, which may be triggered by unpredictable external items.

The urgency of the change is based on how long the implementation can afford to be delayed.

Table 4.6 gives examples of change priorities for corrective changes (fixing identified errors that are hurting the business) and for enhancements (that will deliver additional benefits). Other types of change exist, e.g. to enable continuation of existing benefit, but these two are used to illustrate the concept.

Change planning and scheduling

Careful planning of changes will ensure that there is no ambiguity about what tasks are included in the Change Management process, what tasks are included in other

Table 4.6 Change priority examples

Priority	Corrective change	Enhancement change
Immediate Treat as emergency change (see 4.2.6.9)	Putting life at risk Causing significant loss of revenue or the ability to deliver important public services. Immediate action required	Not appropriate for enhancement changes
High To be given highest priority for change building, testing and implementation resources	Severely affecting some key users, or impacting on a large number of users	Meets legislative requirements Responds to short term market opportunities or public requirements Supports new business initiatives that will increase company market position
Medium	No severe impact, but rectification cannot be deferred until the next scheduled release or upgrade	Maintains business viability Supports planned business initiatives
Low	A change is justified and necessary, but can wait until the next scheduled release or upgrade	Improvements in usability of a service Adds new facilities

processes and how processes interface to any suppliers or projects that are providing a change or release.

Many changes may be grouped into one release and may be designed, tested and released together if the amount of changes involved can be handled by the business, the service provider and its customers. However, if many independent changes are grouped into a release then this may create unnecessary dependencies that are difficult to manage. If not enough changes are grouped into a release then the overhead of managing more releases can be time consuming and waste resources (see paragraph 4.4.5.1 on release and deployment planning).

It is recommended very strongly that Change Management schedule changes to meet business rather than IT needs, e.g. avoiding critical business periods.

Pre-agreed and established change and release windows help an organization improve the planning and throughput of changes and releases. For example a release window in a maintenance period of one hour each week may be sufficient to install minor releases only. Major releases may need to be scheduled with the business and stakeholders at a pre-determined time. This approach is particularly relevant in high change environments where a release is a bottleneck or in high availability services where access to the live systems to implement releases is restricted. In many cases, the change or release may need to be adjusted 'on the fly', and so efficient use of release windows will require:

- A list of possible substitutes to make use of the unexpectedly vacant slot
- Empowerment to make and implement release decisions
- Internal metrics that monitor (and reflect and encourage best use of) change and release windows
- A clear understanding of any sequential dependencies and impact on users.

Wherever possible, Change Management should schedule authorized changes into target release or deployment packages and recommend the allocation of resources accordingly.

Change Management coordinates the production and distribution of a change schedule (CS) and projected service outage (PSO). The SC contains details of all the changes authorized for implementation and their proposed implementation dates. The PSO contains details of changes to agreed SLAs and service availability because of the currently planned SC in addition to planned downtime from other causes such as planned maintenance and data backup. These documents are agreed with the relevant customers within the business, with service level management, with the service desk and with availability management. Once agreed, the service desk should communicate any planned additional downtime to the user community at large, using the most effective methods available.

The latest versions of these documents will be available to stakeholders within the organization, preferably contained within a commonly available internet or intranet server. This can usefully be reinforced with a proactive awareness programme where specific impact can be detected.

Assessing remediation

It is important to develop a remediation plan to address a failing change or release long before implementation. Very often, remediation is the last thing to be considered; risks may be assessed, mitigation plans cast in stone. How to get back to the original start point is often ignored or considered only when regression is the last remaining option.

4.2.6.5 Authorizing the change

Formal authorization is obtained for each change from a change authority that may be a role, person or a group of people. The levels of authorization for a particular type of change should be judged by the type, size or risk of the change, e.g. changes in a large enterprise that affect several distributed sites may need to be authorized by a higher-level change authority such as a global CAB or the Board of Directors.

The culture of the organization dictates, to a large extent, the manner in which changes are authorized. Hierarchical structures may well impose many levels of change authorization, while flatter structures may allow a more streamlined approach.

A degree of delegated authority may well exist within an authorization level, e.g. delegating authority to a change manager according to pre-set parameters relating to:

- Anticipated business risk
- Financial implications
- Scope of the change (e.g. internal effects only, within the finance service, specific outsourced services).

An example of a change authorization hierarchy is shown in Figure 4.5.

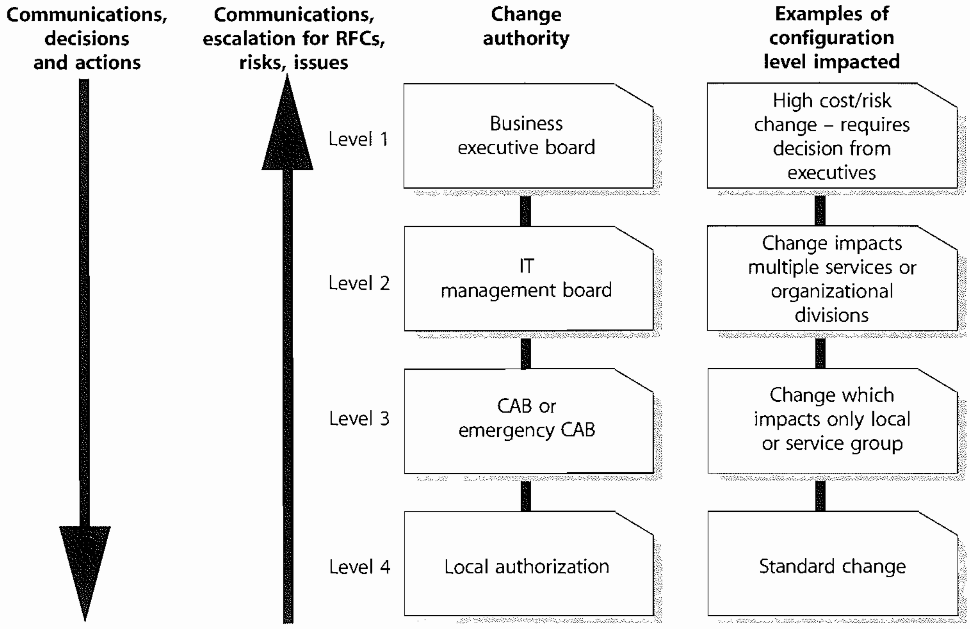

Communications, decisions and actions
Communications, escalation for RFCs, risks, issues
Change authority
Examples of configuration level impacted
Level 1
Business executive board
High cost/risk change – requires decision from executives
Level 2
IT management board
Change impacts multiple services or organizational divisions
Level 3
CAB or emergency CAB
Change which impacts only local or service group
Level 4
Local authorization
Standard change

Figure 4.5 Example of a change authorization model

If change assessment at levels 2, 3, or 4 detects higher levels of risk, the authorization request is escalated to the appropriate higher level for the assessed level of risk. The use of delegated authority from higher levels to local levels must be accompanied by trust in the judgement, access to the appropriate information and supported by management. The level at which change is authorized should rest where accountability for accepting risk and remediation exist.

Should disputes arise over change authorization or rejection, there should be a right of appeal to the higher level.

4.2.6.6 Coordinating change implementation

Authorized RFCs should be passed to the relevant technical groups for building of the changes. It is best practice to do this in a formal way that can be tracked, e.g. using work orders. Building of changes is considered in section 4.4.5.3.

Change Management has responsibility for ensuring that changes are implemented as scheduled. This is largely a coordination role as the actual implementation will be the responsibility of others (e.g. hardware technical specialists will implement hardware changes).

Remediation procedures should be prepared and documented in advance, for each authorized change, so that if errors occur during or after implementation, these procedures can be quickly activated with minimum impact on service quality. Authority and responsibility for invoking remediation is specifically mentioned in change documentation.

Change Management has an oversight role to ensure that all changes that can be are thoroughly tested. In all cases involving changes that have not been fully tested, special care needs to be taken during implementation.

Testing may continue in parallel with early live usage of a service – looking at unusual, unexpected or future situations so that further correcting action can be taken before any detected errors become apparent in live operation.

The implementation of such changes should be scheduled when the least impact on live services is likely. Support staff should be on hand to deal quickly with any incidents that might arise.

4.2.6.7 Review and close change record

On completion of the change, the results should be reported for evaluation to those responsible for managing changes, and then presented as a completed change for stakeholder agreement (including the closing of related incidents, problems or known errors). Clearly, for major changes there will be more customer and stakeholder input throughout the entire process.

A review should also include any incidents arising as a result of the change (if they are known at this stage). If the change is part of a service managed by an external provider, details of any contractual service targets will be required (e.g. no priority 1 incidents during first week after implementation).

A change review (e.g. post-implementation review, PIR) should be carried out to confirm that the change has met its objectives, that the initiator and stakeholders are happy with the results; and that there have been no unexpected side-effects. Lessons learned should be fed back into future changes. Small organizations may opt to use spot checking of changes rather than large-scale PIR; in larger organizations, sampling will have a value when there are many similar changes taking place.

There is a significantly different approach and profile between:

- The review of a service change – immediately visible to the customer and scheduled for discussion at the next service level management review meeting
- An infrastructure change – concerned with how IT delivers rather than what IT delivers, which will be (almost) invisible to the customer.

Change Management must review new or changed services after a predefined period has elapsed. This process will involve CAB members, since change reviews are a standard CAB agenda item. The purpose of such reviews is to establish that:

- The change has had the desired effect and met its objectives
- Users, customers and other stakeholders are content with the results, or to identify any shortcomings
- There are no unexpected or undesirable side-effects to functionality, service levels, warranties, e.g. availability, capacity, security, performance and costs
- The resources used to implement the change were as planned
- The release and deployment plan worked correctly (so include comments from the implementers)
- The change was implemented on time and to cost
- The remediation plan functioned correctly, if needed.

Further details of performing a formal evaluation are provided in Section 4.6. Any problems and discrepancies should be fed back to CAB members (where they have been consulted or where a committee was convened), impact assessors, product authorities and release authorities, so as to improve the processes for the future.

Where a change has not achieved its objectives, Change Management (or the CAB) should decide what follow-up action is required, which could involve raising a revised RFC. If the review is satisfactory or the original change is abandoned (e.g. the circumstances that required the change are no longer current and the requirement disappears) the RFC should be formally closed in the logging system.

4.2.6.8 Change Advisory Board

The Change Advisory Board (CAB) is a body that exists to support the authorization of changes and to assist Change Management in the assessment and prioritization of changes. As and when a CAB is convened, members should be chosen who are capable of ensuring that all changes within the scope of the CAB are adequately assessed from both a business and a technical viewpoint.

The CAB may be asked to consider and recommend the adoption or rejection of changes appropriate for higher-level authorization and then recommendations will be submitted to the appropriate change authority.

To achieve this, the CAB needs to include people with a clear understanding across the whole range of stakeholder needs. The change manager will normally chair the CAB, and potential members include:

- Customer(s)
- User manager(s)
- User group representative(s)
- Applications developers/maintainers
- Specialists/technical consultants
- Services and operations staff, e.g. service desk, test management, ITSCM, security, capacity
- Facilities/office services staff (where changes may affect moves/accommodation and vice versa)
- Contractor's or third parties' representatives, e.g. in outsourcing situations
- Other parties as applicable to specific circumstances (e.g. police if traffic disruptions likely, marketing if public products affected).

It is important to emphasize that the CAB:

- Will be composed according to the changes being considered
- May vary considerably in make-up even across the range of a single meeting
- Should involve suppliers when that would be useful
- Should reflect both users' and customers' views
- Is likely to include the problem manager and service level manager and customer relations staff for at least part of the time.

When the need for emergency change arises, i.e. there may not be time to convene the full CAB, it is necessary to identify a smaller organization with authority to make emergency decisions. This body is the Emergency Change

Advisory Board (ECAB). Change procedures should specify how the composition of the CAB and ECAB will be determined in each instance, based on the criteria listed above and any other criteria that may be appropriate to the business. This is intended to ensure that the composition of the CAB will be flexible, in order to represent business interests properly when major changes are proposed. It will also ensure that the composition of the ECAB will provide the ability, both from a business perspective and from a technical standpoint, to make appropriate decisions in any conceivable eventuality.

A practical tip worth bearing in mind is that the CAB should have stated and agreed evaluation criteria. This will assist in the change assessment activities, acting as a template or framework by which members can assess each change.

CAB meetings

Many organizations are running CABs electronically without frequent face-to-face meetings. There are benefits and problems from such an approach. Much of the assessment and referral activities can be handled electronically via support tools or e-mail. In complex, high-risk or high-impact cases, formal CAB meetings may be necessary.

Handling electronically is more convenient time-wise for CAB members but is also highly inefficient when questions or concerns are raised such that many communications go back and forth. A face-to-face meeting is generally more efficient, but poses scheduling and time conflicts among CAB members as well as significant travel and staff costs for widely dispersed organizations.

Practical experience shows that regular meetings combined with electronic automation is a viable approach for many organizations, and that it can be beneficial to schedule a regular meeting, or when major projects are due to deliver releases. The meetings can then be used to provide a formal review and sign-off of authorized changes, a review of outstanding changes, and, of course, to discuss any impending major changes. Where meetings are appropriate, they should have a standard agenda.

A standard CAB agenda should include:

- Failed changes, unauthorized, backed-out changes, or changes applied without reference to the CAB by incident management, problem management or Change Management
- RFCs to be assessed by CAB members – in structured and priority order
- RFCs that have been assessed by CAB members
- Scheduling of changes and update of change schedule (CS) and PSO
- Change reviews
- The Change Management process, including any amendments made to it during the period under discussion, as well as proposed changes
- Change Management wins/accomplishments for the period under discussion, i.e. a review of the business benefits accrued by way of the Change Management process
- Outstanding changes and changes in progress
- Advance notice of RFCs expected for review at next CAB
- Review of unauthorized changes detected through Configuration Management.

CAB meetings represent a potentially large overhead on the time of members. Therefore all RFCs, together with the SC and PSO, should be circulated in advance, and flexibility allowed to CAB members on whether to attend in person, to send a deputy, or to send any comments. Relevant papers should be circulated in advance to allow CAB members (and others who are required by Change Management or CAB members) to conduct impact and resource assessments.

In some circumstances it will be desirable to table RFCs at one CAB meeting for more detailed explanation or clarification before CAB members take the papers away for consideration, in time for a later meeting. A 'walkthrough' of major changes may be included at a CAB meeting before formal submission of the RFC.

CAB members should come to meetings prepared and empowered to express views and make decisions on behalf of the area they represent in respect of the submitted RFCs, based on prior assessment of the RFCs.

The CAB should be informed of any emergency changes or changes that have been implemented as a workaround to incidents and should be given the opportunity to recommend follow-up action to them.

Note that the CAB is an advisory body only. If the CAB cannot agree to a recommendation, the final decision on whether to authorize changes, and commit to the expense involved, is the responsibility of management (normally the director of IT or the services director, service manager or change manager as their delegated representative). The Change Management authorization plan should specifically name the person(s) authorized to sign off RFCs.

4.2.6.9 Emergency changes

Emergency changes are sometimes required and should be designed carefully and tested before use or the impact of the emergency change may be greater than the original incident. Emergency changes may document some details retrospectively.

The number of emergency changes proposed should be kept to an absolute minimum, because they are generally more disruptive and prone to failure. All changes likely to be required should, in general, be foreseen and planned, bearing in mind the availability of resources to build and test the changes. Nevertheless, occasions will occur when emergency changes are essential and so procedures should be devised to deal with them quickly, without sacrificing normal management controls.

Emergency change is reserved for changes intended to repair an error in an IT service that is negatively impacting the business to a high degree. Changes intended to introduce immediately required business improvements are handled as normal changes, assessed as having the highest urgency.

Emergency change authorization

Defined authorization levels will exist for an emergency change, and the levels of delegated authority must be clearly documented and understood. In an emergency situation it may not be possible to convene a full CAB meeting. Where CAB approval is required, this will be provided by the Emergency CAB (ECAB).

Not all emergency changes will require the ECAB involvement; many may be predictable both in occurrence and resolution and well understood changes available, with authority delegated, e.g. to Operations teams who will action, document and report on the emergency change.

Emergency change building, testing and implementation

Authorized changes are allocated to the relevant technical group for building. Where timescales demand it, Change Management, in collaboration with the appropriate technical manager, ensures that sufficient staff and resources (machine time etc.) are available to do this work. Procedures and agreements – approved and supported by management – must be in place to allow for this. Remediation must also be addressed.

As much testing of the emergency change as is possible should be carried out. Completely untested changes should not be implemented if at all avoidable. Clearly, if a change goes wrong, the cost is usually greater than that of adequate testing. Consideration should be given to how much it would cost to test all changes fully against the cost of the change failing factored by the anticipated likelihood of its failure.

This means that the less a change is considered likely to fail, the more reasonable it may be to reduce the degree of testing in an emergency. (Remember that there is still merit in testing even after a change has gone live.) When only limited testing is possible – and presuming that parallel development of more robust versions continues alongside the emergency change – then testing should be targeted towards:

- Aspects of the service that will be used immediately (e.g. daily entry features, not end-of-month routines)
- Elements that would cause most short-term inconvenience.

The business should be made aware of associated risks and be responsible for ultimately accepting or rejecting the change based on the information presented.

Change Management will give as much advance warning as possible to the service desk and other stakeholders, and arrange for adequate technical presence to be available, to support Service Operations.

If a Change, once implemented, fails to rectify the urgent outstanding error, there may need to be iterative attempts at fixes. Change Management should take responsibility at this point to ensure that business needs remain the primary concern and that each iteration is controlled in the manner described in this section. Change Management should ensure abortive changes are swiftly backed out.

If too many attempts at an emergency change are abortive, the following questions should be asked:

- Has the error been correctly identified, analysed and diagnosed?
- Has the proposed resolution been adequately tested?
- Has the solution been correctly implemented?

In such circumstances, it may be better to provide a partial service, with some user facilities withdrawn, in order to allow the change to be thoroughly tested or to suspend the service temporarily and then implement the change.

Emergency change documentation

It may not be possible to update all Change Management records at the time that urgent actions are being completed (e.g. during overnight or weekend working). It is, however, essential that temporary records are made during such periods, and that all records are completed retrospectively, at the earliest possible opportunity.

Incident control staff, computer operations and network management staff may have delegated authority to circumvent certain types of incident (e.g. hardware failure) without prior authorization by Change Management. Such circumventions should be limited to actions that do not change the specification of service assets and that do not attempt to correct software errors. The preferred methods for circumventing incidents caused by software errors should be to revert to the previous trusted state or version, as relevant, rather than attempting an unplanned and potentially dangerous change. Change approval is still a prerequisite.

Effectively, the emergency change procedure will follow the normal change procedure except that:

- Approval will be given by the ECAB rather than waiting for a CAB meeting
- Testing may be reduced, or in extreme cases forgone completely, if considered a necessary risk to deliver the change immediately
- Documentation, i.e. updating the change record and configuration data, may be deferred, typically until normal working hours.

4.2.7 Triggers, input and output, and inter-process interfaces

Requests for change can be triggered throughout the service lifecycle and at the interfaces with other organizations, e.g. customers and suppliers. There will also be other stakeholders such as partners that may be involved with the Change Management processes.

Typical examples of types of change that trigger the Change Management process are described below.

Strategic changes

Service strategies require changes to be implemented to achieve specific objectives while minimizing costs and risks. There are no cost-free and risk-free strategic plans or initiatives. There are always costs and risks associated with decisions such as introducing new services, entering new market spaces, and serving new customers. The following are examples of programmes and initiatives that implement strategic changes:

- Legal/regulatory change
- Organizational change
- Policy and standards change
- Change after analysing business, customer and user activity patterns
- Addition of new service to the market space
- Updates to the Service Portfolio, customer portfolio or contract portfolio
- Change of sourcing model
- Technology innovation.

Change to one or more services

Changes to the planned services (in the Service Portfolio) and changes to the services in the service catalogue will trigger the Change Management process. These include changes to:

- Service catalogue
- Service package
- Service definition and characteristics
- Release package
- Capacity and resource requirements
- Service level requirements
- Warranties
- Utilities
- Cost of utilization
- Service assets
- Acceptance Criteria
- Predicted quality of service
- Predicted performance
- Predicted value
- Organizational design
- Stakeholder and communications plans
- Physical change in the environment, e.g. building
- Measurement system
- Plans, e.g. capacity, ITSCM, change, transition, test, release and deployment plans
- Decommission/retire services
- Procedures, manuals, service desk scripts.

Operational change

It is important to know the distinction between different types of requests that will be initiated by users. These types of request will depend on the nature of the organization and services and may include requests such as password reset, access request or request to move an IT asset.

Service Operations staff will also implement corrective and preventative changes, via the standard change procedure, that should be managed through Change Management, e.g. server re-boot, which may impact a shared service.

Changes to deliver continual improvement

When CSI determines that an improvement to a Service is warranted, an RFC should be submitted to Change

Management. Changes such as changes to processes can have an effect on service provision and may also affect other CSI initiatives.

Some strategy and service changes will be initiated by CSI.

4.2.7.1 Inputs

Changes may be submitted as an RFC, often with an associated change proposal that provides the detail of how the change will happen, e.g. approach to implementing a legislative change. The change proposal will be based on a change model and will provide more detail about the specific change proposed. The inputs include:

- Policy and strategies for change and release
- Request for Change
- Change proposal
- Plans – change, transition, release, deployment, test, evaluation and remediation
- Current change schedule and PSO
- Current assets or configuration items, e.g. baseline, service package, release package
- As-planned configuration baseline
- Test results, test report and evaluation report.

4.2.7.2 Outputs

Outputs from the process will be:

- Rejected RFCs
- Approved RFCs
- Change to the services, service or infrastructure resulting from approved RFCs
- New, changed or disposed assets or configuration items, e.g. baseline, service package, release package
- Change schedule
- Revised PSO
- Authorized change plans
- Change decisions and actions
- Change documents and records
- Change Management reports.

4.2.7.3 Interfaces

In order to be able to define clear boundaries, dependencies and rules, change and release management should be integrated with processes used for organizational programmes or projects, supplier management and also integrated with suppliers' processes and procedures. There will be occasions when a proposed change will potentially have a wider impact on other parts of the organization (e.g. facilities or business operations), or vice versa, and the service change process must interface appropriately with other processes involved.

Integration with business change processes

Where appropriate, the Change Management should be involved with business programme and business project management teams to ensure that change issues, aims, impacts and developments are exchanged and cascaded throughout the organization where applicable. This means that changes to any business or project deliverables that do not impact services or service components may be subject to business or project Change Management procedures rather than the IT service Change Management procedures. However, care must be taken to ensure that changes to service configuration baselines and releases do follow the Change Management process. The Change Management team will, however, be expected to liaise closely with projects to ensure smooth implementation and consistency within the changing management environments.

Programme and project management

Programme and project management must work in partnership to align all the processes and people involved in service change initiatives. The closer they are aligned, the higher the probability that the change effort will be moved forward for as long as it takes to complete. Change Management representatives may attend relevant Project Board meetings.

Sourcing and partnering arrangements should clearly define the level of autonomy a partner may have in effecting change within their service domain without reference to the overall service provider.

A key component is how deeply change processes and tools are embedded into the supplier organization or vice versa and where the release veto takes place. If the supplier has responsibility for the availability of the operational service, conflicts can arise.

Sourcing and partnering

Sourcing and partnering include internal and external vendors and suppliers who are providing a new or existing service to the organization. Effective Change Management practices and principles must be put into place to manage these relationships effectively to ensure smooth delivery of service. Effort also should be put into finding out how well the partners themselves manage change and choose partner and sourcing relationships accordingly.

It is important to ensure that service providers (outsourced or in house) provide the Change Management function and processes that match the needs of the business and customers. Some organizations in outsourcing situations refer RFCs to their suppliers for estimates prior to approval of changes. For further information, refer to the ITIL Service Design publication and guidance on supplier management.

4.2.7.4 Interfaces within Service Management

The Service Management processes may require change and improvements.

Many will also be involved in the impact assessment and implementation of service changes, as discussed below.

Asset and Configuration Management

The Configuration Management System provides reliable, quick and easy access to accurate configuration information to enable stakeholders and staff to assess the impact of proposed changes and to track changes work flow. This information enables the correct asset and service component versions to be released to the appropriate party or into the correct environment. As changes are implemented, the Configuration Management information is updated.

The CMS may also identify related CI/assets that will be affected by the change, but not included in the original request, or in fact similar CI/assets that would benefit from similar change.

An overview of how the change and Configuration Management processes work together for an individual change is shown in Figure 4.6.

Problem Management

Problem Management is another key process as changes are often required to implement workarounds and to fix known errors. Problem Management is one of the major sources of RFCs and also often a major contributor to CAB discussion.

IT Service Continuity

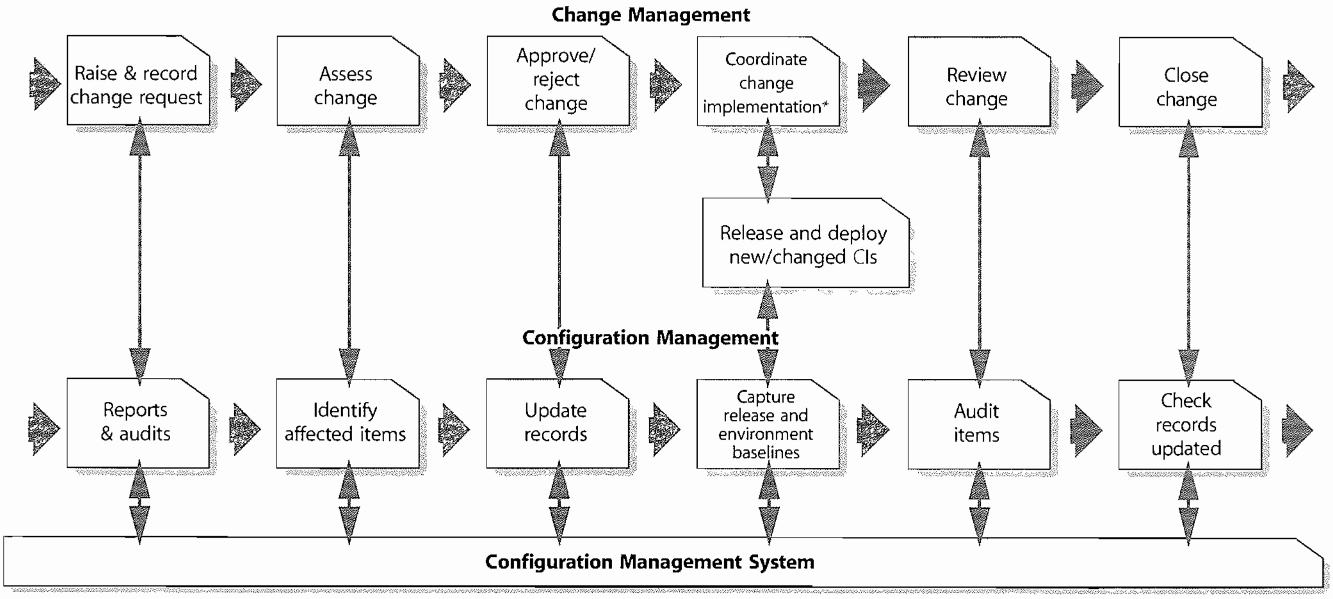
IT Service Continuity has many procedures and plans should be updated via Change Management to ensure that they are accurate, up to date and that stakeholders are aware of changes.

Security Management

Security Management interfaces with Change Management since changes required by security will go via the Change Management process and security will be a key contributor to CAB discussion on many services. Every significant change will be assessed for its potential impact on the security plan.

Capacity and Demand Management

Capacity and Demand Management is a critical aspect of Change Management. Poorly managed demand is a source of costs and risk for service providers because there is always a level of uncertainty associated with the demand for services. Capacity Management has an important role in assessing proposed changes – not only the individual changes but the total impact of changes on service capacity. Changes arising from Capacity Management, including those set out in the capacity plan, will be initiated as RFCs through the change process.



* Includes build and test where applicable

Figure 4.6 Request for Change workflow and key interfaces to Configuration Management

4.2.8 Key performance indicators and metrics

Change Management must ensure that measures have specific meaning. While it is relatively easy to count the number of incidents that eventually generate changes, it is infinitely more valuable to look at the underlying cause of such changes, and to identify trends. Better still to be able to measure the impact of changes and to demonstrate reduced disruption over time because of the introduction of Change Management, and to measure the speed and effectiveness with which the service provider responds to identified business needs.

Measures taken should be linked to business goals wherever practical – and to cost, service availability, and reliability. Any predictions should be compared with actual measurements.

The key performance indicators for Change Management are:

- The number of changes implemented to services which met the customer's agreed requirements, e.g. quality/cost/time (expressed as a percentage of all changes)
- The benefits of change expressed as 'value of improvements made' + 'negative impacts prevented or terminated' compared with the costs of the change process
- Reduction in the number of disruptions to services, defects and re-work caused by inaccurate specification, poor or incomplete impact assessment
- Reduction in the number of unauthorized changes
- Reduction in the backlog of change requests
- Reduction in the number and percentage of unplanned changes and emergency fixes
- Change success rate (percentage of changes deemed successful at review/number of RFCs approved)
- Reduction in the number of changes where remediation is invoked
- Reduction in the number of failed changes
- Average time to implement based on urgency/priority/change type
- Incidents attributable to changes
- Percentage accuracy in change estimate.

Naturally there is other management information required around change and statistics to be gathered and analysed to ensure efficient and effective process, but for organizations with a 'dashboard' reporting approach, these are good metrics to use.

Meaningful measurements are those from which management can make timely and accurate actionable decisions. For example, reporting on the number of changes is meaningless. Reporting on the ratio of changes implemented versus RFCs received provides an efficiency rating. If this rating is low, management can easily see that changes are not being processed in an efficient or effective manner and then take timely action to correct the deficiencies causing this.

4.2.8.1 Examples of the types of measures for change

Some examples of the types of measures used within organizations are listed here; the accrual ones relevant in each different circumstance will vary between organizations and over time, as the change process (and other ITSM elements) mature. Most of the listed measures can be usefully broken down by category, organizational division, geography, supplier, etc.

Output measures

- Number of disruptions, incidents, problems/errors caused by unsuccessful changes and releases
- Inaccurate change specifications (e.g. technical, customer, business)
- Incomplete impact assessment
- Unauthorized business/customer change by business/IT/customer/user asset or configuration item type, e.g. application data
- Percentage reduction in time, effort, cost to make changes and releases (e.g. by service, change type, asset type)
- Service or application re-work caused by inadequate change specification
- Percentage improvement in predictions for time, quality, cost, risk, resource and commercial impact
- Percentage improvement in impact analysis and scheduling of changes safely, efficiently and effectively reduces the risk of changes affecting the live environment
- Percentage reduction in unauthorized changes.

Workloads

- Frequency of change (by service, business area, etc.)
- Volume of change.

Process measures

- People's satisfaction with the speed, clarity, ease of use
- Number and percentage of changes that follow formal Change Management procedures

- Ratio of planned vs unplanned changes (urgent, emergency)
- Ratio of accepted to rejected change requests
- Number of changes recorded and tracked using automated tools
- Time to execute a change (from initiation through each stage in the lifecycle of a change, ending in completion):
  - By lifecycle stage
  - By service
  - By infrastructure platform
- Staff utilization
- Cost against budget.

4.3 SERVICE ASSET AND CONFIGURATION MANAGEMENT

This section addresses the process of Service Asset and Configuration Management (SACM) within IT Service Management. No organization can be fully efficient or effective unless it manages its assets well, particularly those assets that are vital to the running of the customer's or organization's business. This process manages the service assets in order to support the other Service Management processes.

4.3.1 Purpose, goal and objective

The purpose of SACM is to:

- Identify, control, record, report, audit and verify service assets and configuration items, including versions, baselines, constituent components, their attributes, and relationships
- Account for, manage and protect the integrity of service assets and configuration items (and, where appropriate, those of its customers) through the service lifecycle by ensuring that only authorized components are used and only authorized changes are made
- Protect the integrity of service assets and configuration items (and, where appropriate, those of its customers) through the service lifecycle
- Ensure the integrity of the assets and configurations required to control the services and IT infrastructure by establishing and maintaining an accurate and complete Configuration Management System.

The goals of Configuration Management are to:

- Support the business and customer's control objectives and requirements
- Support efficient and effective Service Management processes by providing accurate configuration information to enable people to make decisions at the right time, e.g. to authorize change and releases, resolve incidents and problems faster.
- Minimize the number of quality and compliance issues caused by improper configuration of services and assets
- Optimize the service assets, IT configurations, capabilities and resources.

The objective is to define and control the components of services and infrastructure and maintain accurate configuration information on the historical, planned and current state of the services and infrastructure.

4.3.2 Scope

Asset Management covers service assets across the whole service lifecycle. It provides a complete inventory of assets and who is responsible for their control. It includes:

- Full lifecycle management of IT and service assets, from the point of acquisition through to disposal
- Maintenance of the asset inventory.

Configuration Management ensures that selected components of a complete service, system or product (the configuration) are identified, baselined and maintained and that changes to them are controlled. It also ensures that releases into controlled environments and operational use are done on the basis of formal approvals. It provides a configuration model of the services, assets and infrastructure by recording the relationships between service assets and configuration items. SACM may cover non-IT assets, work products used to develop the services and configuration items required to support the service that are not formally classified as assets.

The scope covers interfaces to internal and external service providers where there are assets and configuration items that need to be controlled, e.g. shared assets.

4.3.3 Value to business

Optimizing the performance of service assets and configurations improves the overall service performance and optimizes the costs and risks caused by poorly managed assets, e.g. service outages, fines, correct licence fees and failed audits.

SACM provides visibility of accurate representations of a service, release, or environment that enables:

- Better forecasting and planning of changes
- Changes and releases to be assessed, planned and delivered successfully
- Incidents and problems to be resolved within the service level targets
- Service levels and warranties to be delivered
- Better adherence to standards, legal and regulatory obligations (less non-conformances)
- More business opportunities as able to demonstrate control of assets and services
- Changes to be traceable from requirements
- The ability to identify the costs for a service.

4.3.4 Policies, principles and basic concepts

In distributed environments and shared services, individual service components exist within many different services and configuration structures. For example, a person may use a desktop computer that is on the network for a building but may be running a central financial system that is linked to a database on the other side of the world. A change to the network or the financial system may have an impact on this person and his/her business process. In web-based services, there may be data feeds and interfaces from and to services owned by other organizations. Changes at these interfaces need to be managed and it is important to identify the interface such as data feeds and the owner/custodian of these. Changes to any interface items need to be managed through Change Management.

4.3.4.1 Service Asset and Configuration Management policies

The first step is to develop and maintain the SACM policies that set the objectives, scope and principles and critical success factors (CSFs) for what is to be achieved by the process. These policies are often considered with the change and Release and Deployment Management policies as they are closely related. The policies will be based on the organization's business drivers, contractual and Service Management requirements and on compliance to applicable laws, regulations and standards.

Asset policies may be applicable for specific asset types or services, e.g. desktop.

There are significant costs and resources implications to implementing SACM and therefore strategic decisions need to be made about the priorities to be addressed. Many IT service providers focus initially on the basic IT assets (hardware and software) and the services and assets that are business critical or covered by legal and regulatory compliance, e.g. Sarbanes-Oxley, software licensing.

Service Asset and Configuration Management principles

The main policy sets out the framework and key principles against which assets and configurations are developed and maintained. Typical principles include:

- Ensuring that Asset and Configuration Management operations costs and resources are commensurate with the potential risks to the services
- The need to deliver corporate governance requirements, e.g. software asset management, Sarbanes-Oxley
- The need to deliver the capability, resources and service warranties as defined by the service level agreements and contracts
- The requirement for available, reliable and cost-effective services
- The requirement for clear economic and performance criteria for interventions that reduce costs or optimize service delivery, e.g. lower maintenance costs
- The application of whole-life cost appraisal methods
- The transformation from 'find and fix' reactive maintenance to 'predict and prevent' proactive management
- The requirement to maintain adequate asset and configuration information for internal and external stakeholders
- The level of control and requirements for traceability and auditability
- The application of continual improvement methods to optimize the service levels, assets and configurations
- Provision of accurate asset and configuration information for other business and Service Management processes
- Integration of Asset and Configuration Management with other processes
- Migration to a common asset and CMS architecture
- Level of automation to reduce errors and costs.

4.3.4.2 Basic concepts

The configuration model

Configuration Management delivers a model of the services, assets and the infrastructure by recording the relationships between configuration items as shown in Figure 4.7. This enables other processes to access valuable information, e.g.:

- To assess the impact and cause of incidents and problems

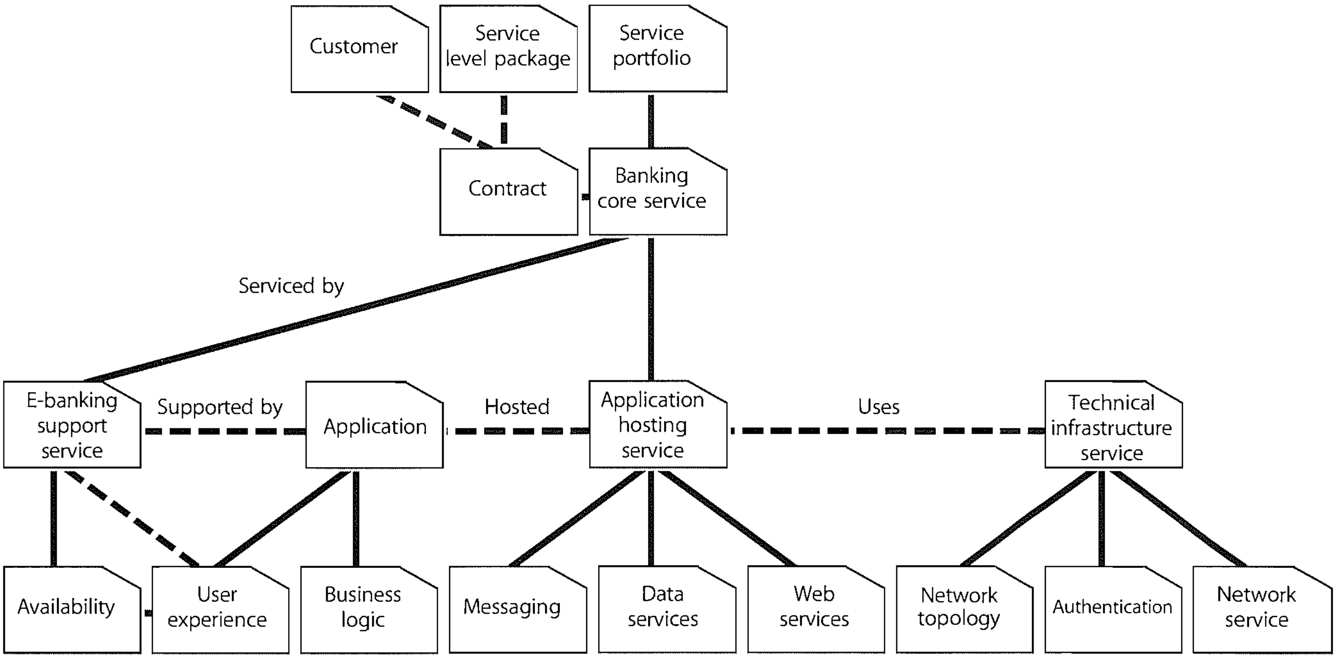


Figure 4.7 Example of a logical configuration model

- To assess the impact of proposed changes
- To plan and design new or changed services
- To plan technology refresh and software upgrades
- To plan release and deployment packages and migrate service assets to different locations and service centres
- To optimize asset utilization and costs, e.g. consolidate data centres, reduce variations and re-use assets.

The real power of Configuration Management's logical model of the services and infrastructure is that it is THE model – a single common representation used by all parts of IT Service Management, and beyond, such as HR, finance, supplier and customers.

'Danish clock'

There is a traditional Danish proverb that runs 'When you have a clock in your house, you know the time – once you get two clocks you are no longer certain.' SACM delivers that one clock for all processes and so glues them together, delivers consistency and helps achieve common purpose. (From Hans Dithmar)

The configuration items and related configuration information can be at varying levels of detail, e.g. an overview of all the services or a detailed level to view the specification for a service component.

Configuration Management should be applied at a more detailed level where the service provider requires tight control, traceability and tight coupling of configuration information through the service lifecycle.

Configuration items

A configuration item (CI) is an asset, service component or other item that is, or will be, under the control of Configuration Management. Configuration items may vary widely in complexity, size and type, ranging from an entire service or system including all hardware, software, documentation and support staff to a single software module or a minor hardware component. Configuration items may be grouped and managed together, e.g. a set of components may be grouped into a release. Configuration items should be selected using established selection criteria, grouped, classified and identified in such a way that they are manageable and traceable throughout the service lifecycle.

There will be a variety of CIs; the following categories may help to identify them.

- **Service lifecycle CIs** such as the Business Case, Service Management Plans, service lifecycle plans, Service Design Package, release and change plans, and test plans. They provide a picture of the service provider's services, how these services will be delivered, what benefits are expected, at what cost, and when they will be realized.
- **Service CIs** such as:
  - Service capability assets: management, organization, processes, knowledge, people
  - Service resource assets: financial capital, systems, applications, information, data, infrastructure and facilities, financial capital, people

- Service model
- Service package
- Release package
- Service acceptance criteria.

- **Organization CIs** – Some documentation will define the characteristics of a CI whereas other documentation will be a CI in its own right and need to be controlled, e.g. the organization's business strategy or other policies that are internal to the organization but independent of the service provider. Regulatory or statutory requirements also form external products that need to be tracked, as do products shared between more than one group.
- **Internal CIs** comprising those delivered by individual projects, including tangible (data centre) and intangible assets such as software that are required to deliver and maintain the service and infrastructure.
- **External CIs** such as external customer requirements and agreements, releases from suppliers or sub-contractors and external services.
- **Interface CIs** that are required to deliver the end-to-end service across a service provider interface (SPI).

4.3.4.3 Configuration Management System

To manage large and complex IT services and infrastructures, Service Asset and Configuration Management requires the use of a supporting system known as the Configuration Management System (CMS).

The CMS holds all the information for CIs within the designated scope. Some of these items will have related specifications or files that contain the contents of the item, e.g. software, document or photograph. For example, a Service CI will include the details such as supplier, cost, purchase date and renewal date for licences and maintenance contracts and the related documentation such as SLAs and underpinning contracts.

The CMS is also used a for wide range of purposes, for example asset data held in the CMS may be made available to external financial Asset Management systems to perform specific Asset Management processes reporting outside of Configuration Management.

The CMS maintains the relationships between all service components and any related incidents, problems, known errors, change and release documentation and may also

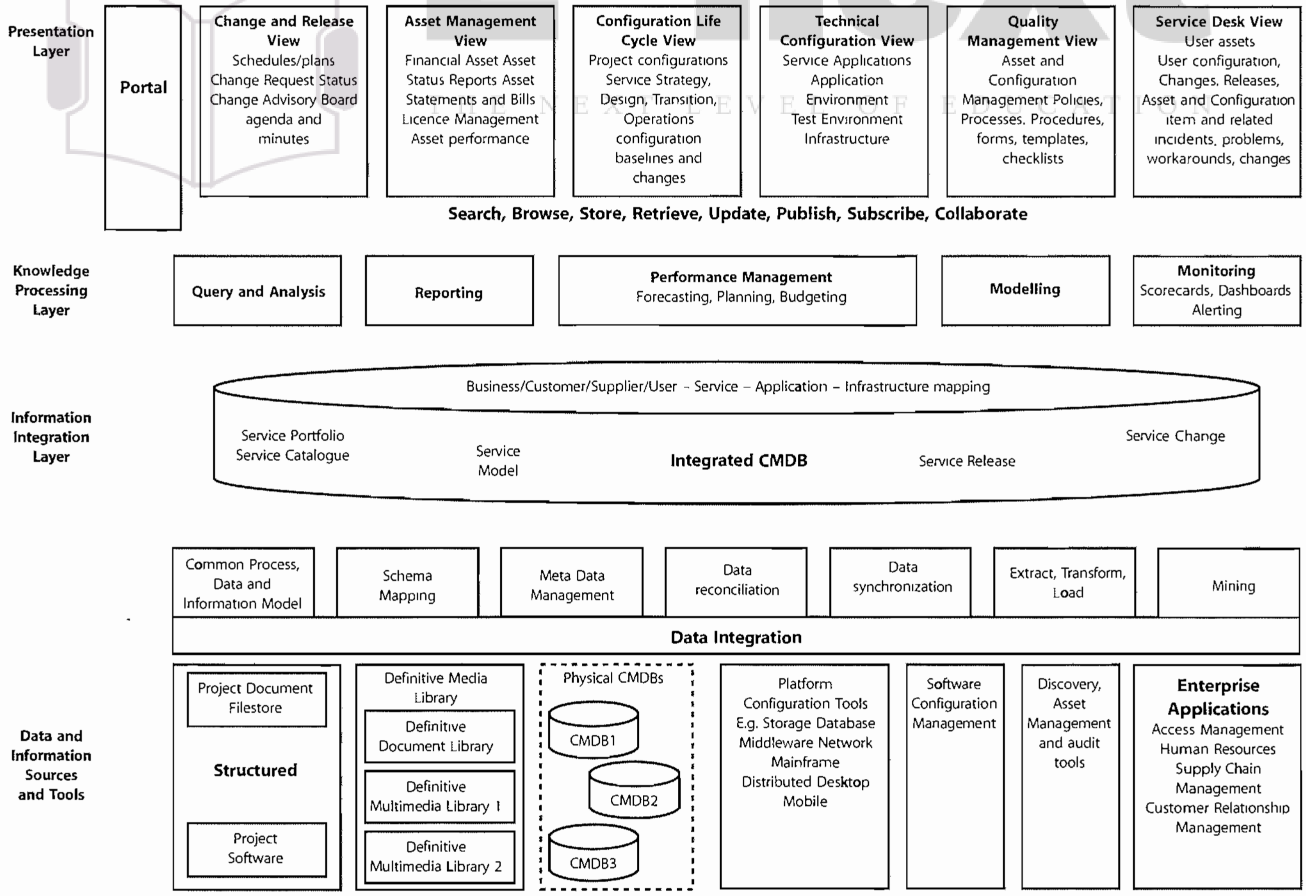

Presentation Layer
Portal
Change and Release View
Schedules/plans Change Request Status Change Advisory Board agenda and minutes
Asset Management View
Financial Asset Asset Status Reports Asset Statements and Bills Licence Management Asset performance
Configuration Life Cycle View
Project configurations Service Strategy, Design, Transition, Operations configuration baselines and changes
Technical Configuration View
Service Applications Application Environment Test Environment Infrastructure
Quality Management View
Asset and Configuration Management Policies, Processes, Procedures, forms, templates, checklists
Service Desk View
User assets User configuration, Changes, Releases, Asset and Configuration item and related incidents, problems, workarounds, changes
Search, Browse, Store, Retrieve, Update, Publish, Subscribe, Collaborate
Knowledge Processing Layer
Query and Analysis
Reporting
Performance Management
Forecasting, Planning, Budgeting
Modelling
Monitoring
Scorecards, Dashboards Alerting
Information Integration Layer
Business/Customer/Supplier/User – Service – Application – Infrastructure mapping
Service Portfolio
Service Catalogue
Service Model
Integrated CMDB
Service Release
Service Change
Common Process, Data and Information Model
Schema Mapping
Meta Data Management
Data reconciliation
Data synchronization
Extract, Transform, Load
Mining
Data Integration
Data and Information Sources and Tools
Project Document Filestore
Structured
Project Software
Definitive Media Library
Definitive Document Library
Definitive Multimedia Library 1
Definitive Multimedia Library 2
Physical CMDBs
CMDB1
CMDB2
CMDB3
Platform Configuration Tools E.g. Storage Database Middleware Network Mainframe Distributed Desktop Mobile
Software Configuration Management
Discovery, Asset Management and audit tools
Enterprise Applications
Access Management Human Resources Supply Chain Management Customer Relationship Management

Figure 4.8 Example of a Configuration Management System

contain corporate data about employees, suppliers, locations and business units, customers and users.

Figure 4.8 shows how the CMS covers the data and information layers of the knowledge/information/ knowledge hierarchy explained in section 4.7, Knowledge Management.

At the data level, the CMS may take data from several physical CMDBs, which together constitute a federated CMDB. Other data sources will also plug into the CMS such as the definitive media libraries. The CMS will provide access to data in asset inventories wherever possible rather than duplicating data.

Example of multiple Configuration Management databases

In the commonly encountered partially outsourced service provider, some elements of the Service Management will be outsourced while others will remain in house, and different elements may be outsourced to different external suppliers. For example the network and hardware support may be handled by supplier A, environment and facilities management by supplier B, and multiple applications suppliers and incident management handled internally. The service desk will access information to assist them from the CMS, but that system will derive its data input from discrete repositories – each one a CMDB – owned and maintained by the three parties but working together to supply a single consistent information set.

Configuration information evolves as the service is developed through the service lifecycle. Often there are separate mechanisms for managing different service lifecycle stages as well as different means of managing different applications and platforms.

The CMS typically contains configuration data and information that combined into an integrated set of views for different stakeholders through the service lifecycle as illustrated in Figure 4.8. It therefore needs to be based on appropriate web, reporting and database technologies that provide flexible and powerful visualization and mapping tools, interrogation and reporting facilities.

Many organizations are already using some elements of SACM, often maintaining records in documents, spreadsheets or local databases, and some of these may be used in the overall CMS.

Automated processes to load and update the Configuration Management database should be developed where possible so as to reduce errors and optimize costs. Discovery tools, inventory and audit tools, enterprise systems and network management tools can be interfaced to the CMS. These tools can be used initially to populate a CMDB, and subsequently to compare the actual 'live' configuration with the information and records stored in the CMS.

Secure libraries and secure stores

A secure library is a collection of software, electronic or document CIs of known type and status. Access to items in a secure library is restricted. Libraries are used for controlling and releasing components throughout the service lifecycle, e.g. in design, building, testing, deployment and operations.

A secure store is a location that warehouses IT assets. It is identified within SACM, e.g. secure stores used for desktop deployment. Secure stores play an important role in the provision of security and continuity – maintaining reliable access to equipment of known quality.

The Definitive Media Library

The Definitive Media Library (DML) is the secure library in which the definitive authorized versions of all media CIs are stored and protected. It stores master copies of versions that have passed quality assurance checks. This library may in reality consist of one or more software libraries or file-storage areas, separate from development, test or live file-store areas. It contains the master copies of all controlled software in an organization. The DML should include definitive copies of purchased software (along with licence documents or information), as well as software developed on site. Master copies of controlled documentation for a system are also stored in the DML in electronic form.

The DML will also include a physical store to hold master copies, e.g. a fireproof safe. Only authorized media should be accepted into the DML, strictly controlled by SACM.

The DML is a foundation for Release and Deployment Management (see section 4.4 on the release and deployment process).

The exact configuration of the DML is defined during the planning activities. The definition includes:

- Medium, physical location, hardware and software to be used, if kept online – some Configuration Management support tools incorporate document or software libraries, which can be regarded as a logical part of a DML
- Naming conventions for filestore areas and physical media
- Environments supported, e.g. test and live environments

- Security arrangements for submitting changes and issuing documentation and software, plus backup and recovery procedures
- The scope of the DML, e.g. source code, object code from controlled builds and associated documentation
- Archive and retention periods
- Capacity plans for the DML and procedures for monitoring growth in size
- Audit procedures
- Procedures to ensure that the DML is protected from erroneous or unauthorized change (e.g. entry and exit criteria for items).

Figure 4.9 shows the relationship between the DML and the CMDB.

Definitive spares

An area should be set aside for the secure storage of definitive hardware spares. These are spare components and assemblies that are maintained at the same level as the comparative systems within the controlled test or live environment. Details of these components, their locations and their respective builds and contents should be comprehensively recorded in the CMS. These can then be used in a controlled manner when needed for additional systems or in the recovery from incidents. Once their (temporary) use has ended, they are returned to the spares store or replacements are obtained.

Configuration baseline

A configuration baseline is the configuration of a service, product or infrastructure that has been formally reviewed and agreed on, that thereafter serves as the basis for further activities and that can be changed only through formal change procedures. It captures the structure, contents and details of a configuration and represents a set of configuration items that are related to each other.

Establishing a baseline provides the ability to:

- Mark a milestone in the development of a service, e.g. Service Design baseline
- Build a service component from a defined set of inputs
- Change or rebuild a specific version at a later date
- Assemble all relevant components in readiness for a change or release
- Provide the basis for a configuration audit and back out, e.g. after a change.

Snapshot

A snapshot of the current state of a configuration item or an environment, e.g. from a discovery tool. This snapshot is recorded in the CMS and remains as a fixed historical record. Sometimes this is referred to a footprint. A snapshot is not necessarily formally reviewed and agreed on – it is just a documentation of a state, which may

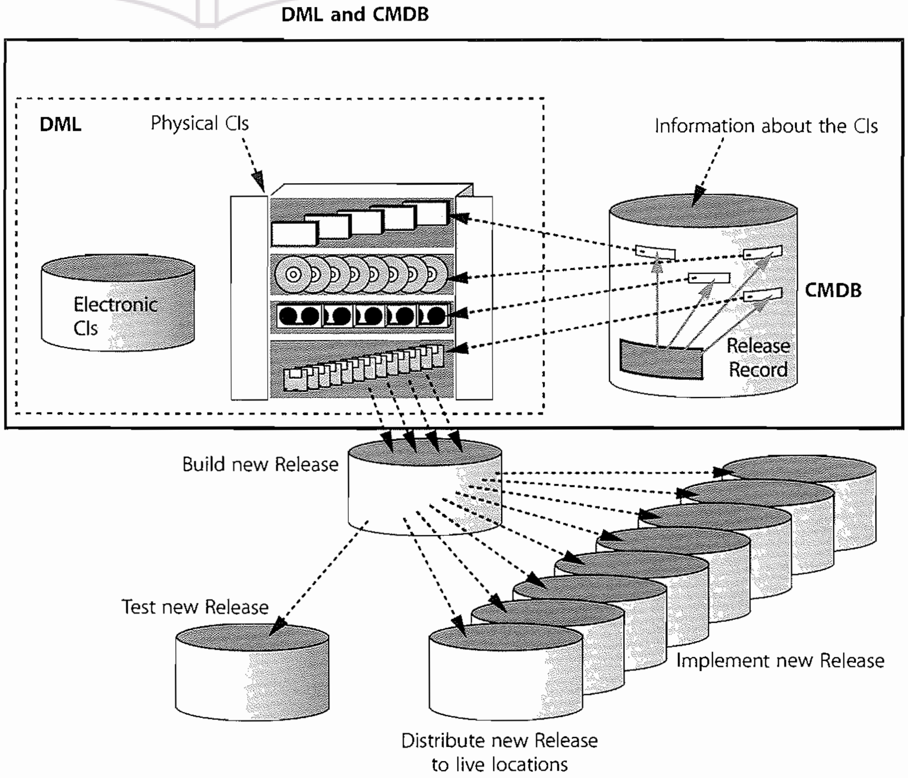


Figure 4.9 The relationship between the Definitive Media Library and the Configuration Management Database

contain faults and unauthorized CIs. One example is where a snapshot is established after an installation, perhaps using a discovery tool, and later compared to the original configuration baseline.

The snapshot:

- Enables problem management to analyse evidence about a situation pertaining at the time incidents actually occurred
- Facilitates system restore to support security scanning software.

4.3.5 Process activities, methods and techniques

4.3.5.1 Asset and Configuration Management activities

High-level activities for Asset and Configuration Management are shown in an example of an activity model in Figure 4.10.

The activity model illustrated in Figure 4.10 is often used where there are many parties or suppliers and activities need to be established to obtain the configuration information and data from third parties.

4.3.5.2 Management and planning

There is no standard template for determining the optimum approach for SACM. The management team and Configuration Management should decide what level of Configuration Management is required for the selected service or project that is delivering changes and how this level will be achieved. This is documented in a Configuration Management Plan. Often there will be a Configuration Management Plan for a project, service or groups of services, e.g. network services. These plans define the specific Configuration Management activities within the context of the overarching Service Asset and Configuration Management strategy.

An example of the contents for an Asset or Configuration Management Plan is shown below.

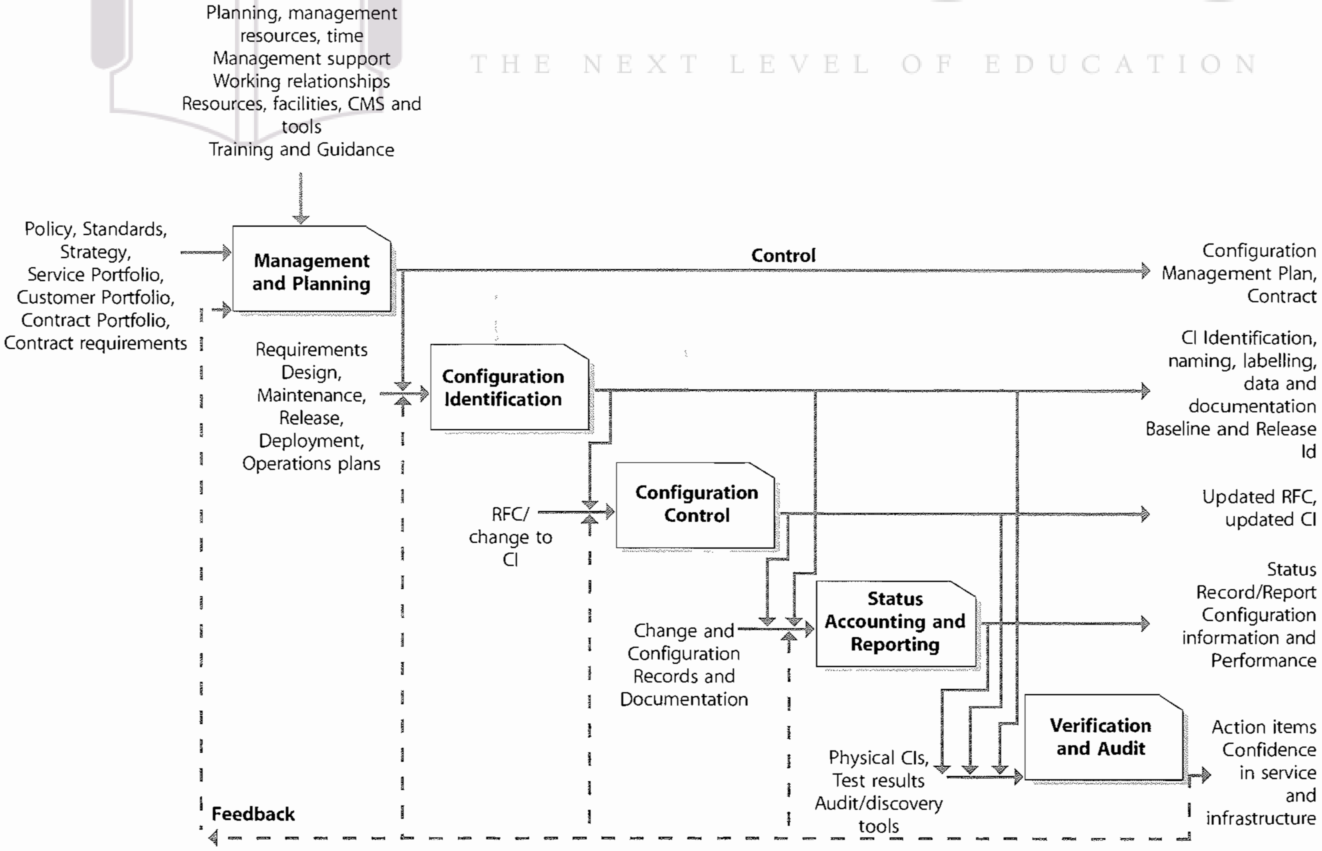

Planning, management resources, time
Management support
Working relationships
Resources, facilities, CMS and tools
Training and Guidance
Policy, Standards, Strategy, Service Portfolio, Customer Portfolio, Contract Portfolio, Contract requirements
Management and Planning
Control
Configuration Management Plan, Contract
Requirements Design, Maintenance, Release, Deployment, Operations plans
Configuration Identification
CI Identification, naming, labelling, data and documentation Baseline and Release Id
RFC/ change to CI
Configuration Control
Updated RFC, updated CI
Change and Configuration Records and Documentation
Status Accounting and Reporting
Status Record/Report Configuration information and Performance
Physical CIs, Test results Audit/discovery tools
Verification and Audit
Action items Confidence in service and infrastructure
Feedback

Figure 4.10 Typical Configuration Management activity model

Example of Asset and Configuration Management Plan contents

Context and purpose.

Scope:

- Applicable services
- Environments and infrastructure
- Geographical locations.

Requirements:

- Link to policy, strategy
- Link to business, Service Management and contractual requirements
- Summarize requirements for accountability, traceability, auditability
- Link to requirements for the Configuration Management System (CMS).

Applicable policies and standards:

- Policies
- Industry standards, e.g. ISO/IEC 20000, ISO/IEC 19770-1
- Internal standards relevant to Configuration Management, e.g. hardware standards, desktop standards.

Organization for Configuration Management:

- Roles and responsibilities
- Change and configuration control boards
- Authorization – for establishing baseline, changes and releases.

Asset and Configuration Management System and tools.

Selection and application of processes and procedures to implement Asset and Configuration Management activities, e.g.:

- Configuration identification
- Version management
- Interface management
- Supplier management
- Configuration Change Management
- Release and deployment
- Build management
- Establishing and maintaining configuration baselines
- Maintaining the CMS
- Reviewing the integrity of configurations and the CMS (verification and audit).

Reference implementation plan, e.g. data migration and loading, training and knowledge transfer plan.

Relationship management and interface controls, for example:

- With financial Asset Management
- With projects
- With development and testing
- With customers
- With service provider interfaces (SPI)
- With operations including the service desk.

Relationship management and control of suppliers and sub-contractors.

4.3.5.3 Configuration identification

When planning configuration identification it is important to:

- Define how the classes and types of assets and configuration items are to be selected, grouped, classified and defined by appropriate characteristics, e.g. warranties for a service, to ensure that they are manageable and traceable throughout their lifecycle
- Define the approach to identification, uniquely naming and labelling all the assets or service components of interest across the service lifecycle and the relationships between them
- Define the roles and responsibilities of the owner or custodian for configuration item type at each stage of its lifecycle, e.g. the service owner for a service package or release at each stage of the service lifecycle.

The configuration identification process activities are to:

- Define and document criteria for selecting configuration items and the components that compose them
- Select the configuration items and the components that compose them based on documented criteria
- Assign unique identifiers to configuration items
- Specify the relevant attributes of each configuration item
- Specify when each configuration item is placed under Configuration Management
- Identify the owner responsible for each configuration item.

Configuration structures and the selection of configuration items

The configuration model should describe the relationship and position of CIs in each structure. There should be service configuration structures that identify all the components in a particular service (e.g. the retail service).

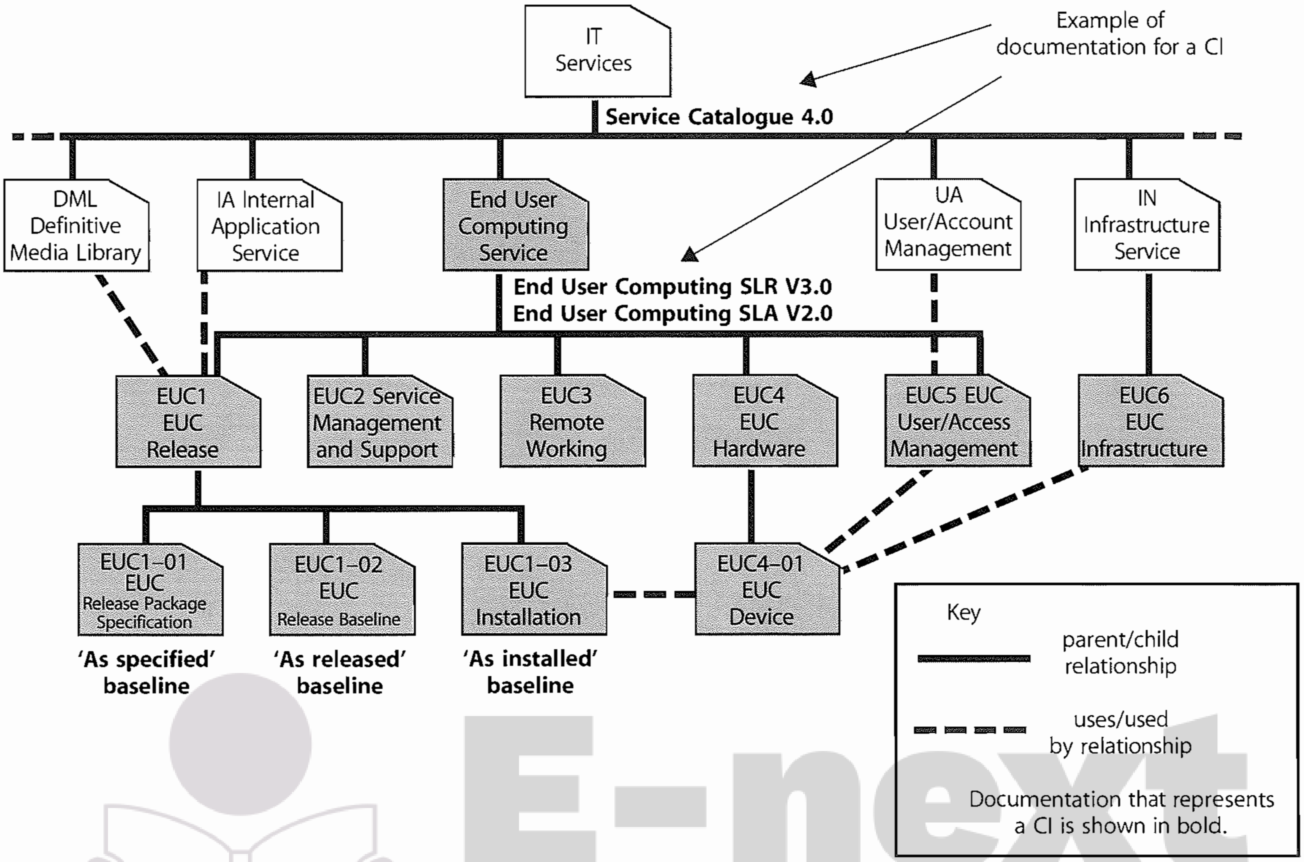

IT Services
Example of documentation for a CI
Service Catalogue 4.0
DML Definitive Media Library
IA Internal Application Service
End User Computing Service
UA User/Account Management
IN Infrastructure Service
End User Computing SLR V3.0
End User Computing SLA V2.0
EUC1 EUC Release
EUC2 Service Management and Support
EUC3 Remote Working
EUC4 EUC Hardware
EUC5 EUC User/Access Management
EUC6 EUC Infrastructure
EUC1–01 EUC Release Package Specification
EUC1–02 EUC Release Baseline
EUC1–03 EUC Installation
EUC4–01 EUC Device
'As specified' baseline
'As released' baseline
'As installed' baseline
Key
parent/child relationship
uses/used by relationship
Documentation that represents a CI is shown in bold.

Figure 4.11 (a) Example configuration breakdown for an end-user computing service

An important part of Configuration Management is deciding the level at which control is to be exercised, with top-level CIs broken down into components which are themselves CIs, and so on.

CIs should be selected by applying a top down approach, considering if it is sensible to break down a CI into component CIs. A CI can exist as part of any number of different CIs or CI groups at the same time. For instance, a database product may be used by many applications. Usage links to re-usable and common components of the service should be defined – for instance, a configuration structure for a retail service will use infrastructure CIs such as servers, network and software CIs. The ability to have multiple views through different configuration structures improves accessibility, impact analysis and reporting.

Configuration Management of work products and service components from the service lifecycle may be performed at several levels of granularity. The items placed under Configuration Management will typically include service bundles, service packages, service components, release packages and products that are delivered to the customer, designated internal work products, acquired services, products, tools, systems and other items that are used in creating and describing the configurations required to design, transition and operate the service.

Figure 4.11 (a) and (b) gives an example in schematic representation of how a CI structure for an end-user computing service and a Managed Virtual System might be broken down.

Choosing the right CI level is a matter of achieving a balance between information availability, the right level of control, and the resources and effort needed to support it. Information at a low CI level may not be valuable – for example, although a keyboard is usually exchanged independently, the organization sees it as a consumable, so does not store data about it. CI information is valuable only if it facilitates the management of change, the control of incidents and problems, or the control of assets that can be independently moved, copied or changed.

Factors that influence recording level of configuration items

The factors that affect choice of lowest CI level are not just financial. As mentioned above most organizations do not store data on keyboards, because they consider them consumables, to be thrown away when not working, as one would a broken pen. However, some organizations find it

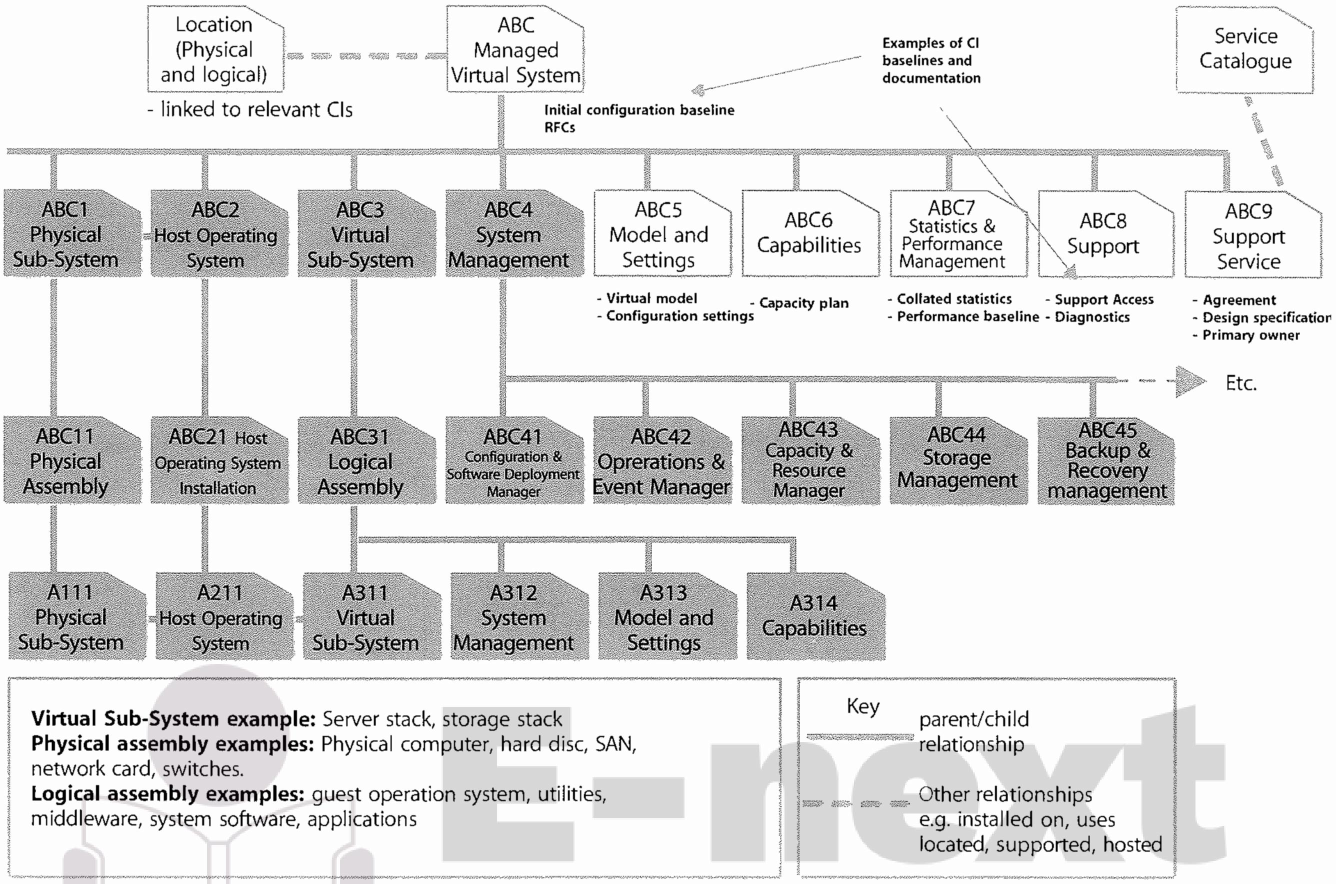

Location (Physical and logical)
- linked to relevant CIs
ABC Managed Virtual System
Initial configuration baseline
RFCs
Examples of CI baselines and documentation
Service Catalogue
ABC1 Physical Sub-System
ABC2 Host Operating System
ABC3 Virtual Sub-System
ABC4 System Management
ABC5 Model and Settings
- Virtual model
- Configuration settings
ABC6 Capabilities
- Capacity plan
ABC7 Statistics & Performance Management
- Collated statistics
- Performance baseline
ABC8 Support
- Support Access
- Diagnostics
ABC9 Support Service
- Agreement
- Design specification
- Primary owner
Etc.
ABC11 Physical Assembly
ABC21 Host Operating System Installation
ABC31 Logical Assembly
ABC41 Configuration & Software Deployment Manager
ABC42 Oprerations & Event Manager
ABC43 Capacity & Resource Manager
ABC44 Storage Management
ABC45 Backup & Recovery management
A111 Physical Sub-System
A211 Host Operating System
A311 Virtual Sub-System
A312 System Management
A313 Model and Settings
A314 Capabilities
Virtual Sub-System example: Server stack, storage stack
Physical assembly examples: Physical computer, hard disc, SAN, network card, switches.
Logical assembly examples: guest operation system, utilities, middleware, system software, applications
Key
parent/child relationship
Other relationships e.g. installed on, uses located, supported, hosted

Figure 4.11 (b) Example configuration breakdown for a Managed Virtual System

worth retaining data on keyboards – for example in the United Nations, which supports many different languages within its office building, recording the specific language keyboard used is an important factor in speedy incident resolution when keyboards fail, i.e. they know which kind of replacement keyboard to send to any given user.

The organization should plan to review the CI level regularly – to confirm (or otherwise) that information down to a low level is still valuable and useful, and that the handling of changes and problems and the management of assets are not deficient because the CMDB does not go down to a sufficiently low level.

Each asset and CI needs to be uniquely identified, whether it is generated inside or outside the organization. The identification should also differentiate between successive versions and should enable the items under control to be unambiguously traceable to their specifications or equivalent, documented descriptions. Configuration descriptions and data should conform, where possible, to service, product or technology standards. Configuration data should permit forward and backward traceability to other baselined configuration states, where required.

Naming configuration items

Naming conventions should be established and applied to the identification of CIs, configuration documents and changes, as well as to baselines, builds, releases and assemblies.

Individual CIs should be uniquely identifiable by means of the identifier and version. The version identifies an updated instance of what can be regarded as the same CI. More than one version of a CI can coexist at any given time. The naming conventions should be unique and take into account the existing corporate or supplier naming/numbering structures. The naming conventions or information management system should include the management of:

- Hierarchical relationships between CIs within a configuration structure
- Hierarchical or subordinate relationships in each CI
- Relationships between CIs and their associated documents
- Relationships between CIs and changes
- Relationships between CIs, incidents, problems and known errors.

Configuration Management should arrange for a naming convention to be established for all documents, e.g. RFCs. Document templates are a good method for standardizing configuration documentation. Without templates there are often far too many documents generated with overlapping content that can make executing changes extremely difficult.

Each type of template and form should be uniquely identifiable with a version number. A typical method of identification is <Form type>_nnnn where nnnn is a sequentially assigned number for each new instance of the form.

When the naming convention is being planned, it is very important that sufficient account is taken of possible future growth. Identifiers should be relatively short, but meaningful, and should follow existing conventions wherever possible. For hardware, if the CI naming conventions are not based on suppliers' device names and models, a mechanism should be set up to relate Configuration Management and suppliers' identifiers to each other, for example, for the convenience of procurement staff and hardware engineers. Standard terminology and abbreviations should be used throughout the organization as far as possible (e.g. NYC rather than sometimes NY or N York). Failure to do so will result in an inability to match common incidents, problems etc. Attributes that might change should never be used as a part of CI naming.

Labelling configuration items

All physical device CIs should be labelled with the configuration identifier so that they can be easily identified. Plans should be made to label CIs and to maintain the accuracy of their labels.

Items need to be distinguished by unique, durable identification, e.g. labels or markings that follow relevant standards where appropriate. Physical non-removable asset tags (labels) should be attached to all hardware CIs; cables/lines should be clearly labelled at each end and at any inspection points. It is advisable to use a standard format and colour for all such labels, because this makes it easier for users to identify and quote from them, for instance when telephoning the service desk to report a fault. Barcode-readable labels improve the efficiency of physical audits. A standard policy on labelling hardware is similarly beneficial at the service desk, e.g. if all hardware is labelled in the bottom left-hand corner of the left side, it is much quicker and easier to explain to the user where they will find the required information.

Attributes for configuration items

Attributes describe the characteristics of a CI that are valuable to record and which will support SACM and the ITSM processes it supports.

The SACM plan references the configuration information and data architecture. This includes the attributes to be recorded for each type of asset or CI. Typical attributes include:

- Unique identifier
- CI type
- Name/description
- Version (e.g. file, build, baseline, release)
- Location
- Supply date
- Licence details, e.g. expiry date
- Owner/custodian
- Status
- Supplier/source
- Related document masters
- Related software masters
- Historical data, e.g. audit trail
- Relationship type
- Applicable SLA.

These attributes will define specific functional and physical characteristics of each type of asset and CI, e.g. size or capacity, together with any documentation or specifications.

Defining configuration documentation

The characteristics of a CI are often contained in documents. For example, the service definition, requirements specification and service level agreement for a service describe the characteristics of a Service CI. Many organizations specify mandatory and optional documents that describe a CI and use document templates to ensure consistent information is entered. Table 4.7 is a RACI (Responsible, Accountable, Consulted, Informed) chart, which illustrates the types of documentation of service assets or configuration items that are the responsibility of different service lifecycle stages and typical documentation.

Collecting CI attribute data can facilitate use/re-use/reference to existing documents, data, files, records, spreadsheets etc. This will help users implementing this to determine a good approach to collecting data.

Table 4.7 Configuration documentation for assets and responsibilities through the service lifecycle

Service lifecycle stage	Examples of service lifecycle assets and CIs impacted	Service Strategy	Service Design	Service Transition	Service Operation	Continual Service Improvement
Service Strategy	Portfolios – service contract, customer Service Strategy requirements Service lifecycle model	A	C	C	R	C
Service Design	Service package (including SLA) Service Design package, e.g. service model, contract, supplier's Service Management Plan, process interface definition, customer engagement plan Release policy Release package definition	I	A	C	R	C
Service Transition	Service Transition model Test plan Controlled environments Build/installation plan Build specification Release plan Deployment plan CMS SKMS Release package Release baseline Release documentation Evaluation report Test report	I	C	A	R	C
Service Operations	Service Operations model Service support model Service desk User assets User documentation Operations documentation Support documentation	I	C	C	A/R	R
Continual Service Improvement	CSI model Service improvement plan Service reporting process	A/C	A/C	A/C	R	A

R=Responsible, A=Accountable, C=Consulted, I=Informed

Relationships

Relationships describe how the configuration items work together to deliver the services. These relationships are held in the CMS – this is the major difference between what is recorded in a CMS and what is held in an asset register.

The relationships between CIs are maintained so as to provide dependency information. For example:

- A CI is a part of another CI, e.g. a software module is part of a program; a server is part of a site infrastructure – this is a 'parent–child' relationship.
- A CI is connected to another CI, e.g. a desktop computer is connected to a LAN.
- A CI uses another CI, e.g. a program uses a module from another program; a business service uses an infrastructure server.
- A CI is installed on another, e.g. MS Project is installed on a desktop PC.

Although a 'child' CI should be 'owned' by one 'parent' CI, it can be 'used by' any number of other CIs. If a standard desktop build is supplied and installed on all PCs within a division or location, then that build, including all the software CIs included, will be a CI that is linked by a relationship to the PCs. The software included will be 'part of' the build. This can considerably reduce the number of relationships that are needed, compared with when individual software CIs relationships are used.

Relationships are also the mechanism for associating RFCs, incident records, problem records, known errors and release records with the services and IT infrastructure CIs to which they refer. All these relationships should be included in the CMS. RFCs and change and release records will identify the CIs affected.

Some of these relationships were shown in Figure 4.11. For example, EUC is the parent CI of EUC1 to EUC5 and EUC1 is in turn the parent of three CIs, EUC1_01 to EUC1_03, shown as the next level in the hierarchy. EUC1 uses the DML and Internal Application (IA) service.

Relationships may be one-to-one, one-to-many and many-to-one. Placing portfolios under the control of the CMS provides a good example. The combination of Service Portfolios and customer portfolios generates the contract portfolio. In other words, every item in the contract portfolio is mapped to at least one item in the Service Portfolio and at least one item in the customer portfolio.

Types of configuration item

Components should be classified into asset or CI types because this helps to identify and document what is in use, the status of the items and where they are located. Typical CI types include service, hardware, software, documentation and staff.

Identification of media libraries

Physical and electronic media libraries should be uniquely identified and recorded in the CMS with the following information:

- Contents, location and medium of each library
- Conditions for entering an item, including the minimum status compatible with the contents of the library
- How to protect the libraries from malicious and accidental harm and deterioration, together with effective recovery procedures
- Conditions and access controls for groups or types of person registering, reading, updating, copying, removing and deleting CIs
- Scope of applicability, e.g. applicable from environment 'system test' through to 'operation'.

Identification of configuration baselines

Configuration baselines should be established by formal agreement at specific points in time and used as departure points for the formal control of a configuration. Configuration baselines plus approved changes to those baselines together constitute the currently approved configuration. Specific examples of baselines that may be identified include:

- A particular 'standard' CI needed when buying many items of the same type (e.g. desktop computer) over a protracted period; if some are to include additional components (e.g. a DVD writer), this could correspond to 'baseline plus'; if all future desktop computers are to have features then a new baseline is created
- An application release and its associated documentation.

Several baselines corresponding to different stages in the life of a 'baselined item' can exist at any given time – for example, the baseline for an application release that is currently live, the one that was last live and has now been archived, the one that will next be installed (subject to change under Configuration Management control), and one or more under test. Furthermore, if, for instance, new software is being introduced gradually regionally, more than one version of a baseline could be 'live' at the same time. It is therefore best to refer to each by a unique version number, rather than 'live', 'next' or 'old'.

By consolidating the evolving configuration states of configuration items to form documented baselines at designated points or times the Configuration Management will be more effective and efficient. Each baseline is a mutually consistent set of CIs that can be declared at key milestones. An example of a baseline is an approved description of a service that includes internally consistent versions of requirements, requirement traceability matrices, design, specific service components and user documentation.

Each baseline forms a frame of reference for the service lifecycle as a whole. Baselines provide the basis for assessing progress and undertaking further work that is internally self-consistent and stable. For example, the Service Portfolio and the Business Case for a Service should present a consistent and clear definition of what the service package is intending to deliver. This may form the 'scope baseline' for the service(s) and give internal and external parties a clear basis for subsequent analysis and development. An example of the baseline points is shown in Figure 4.12.

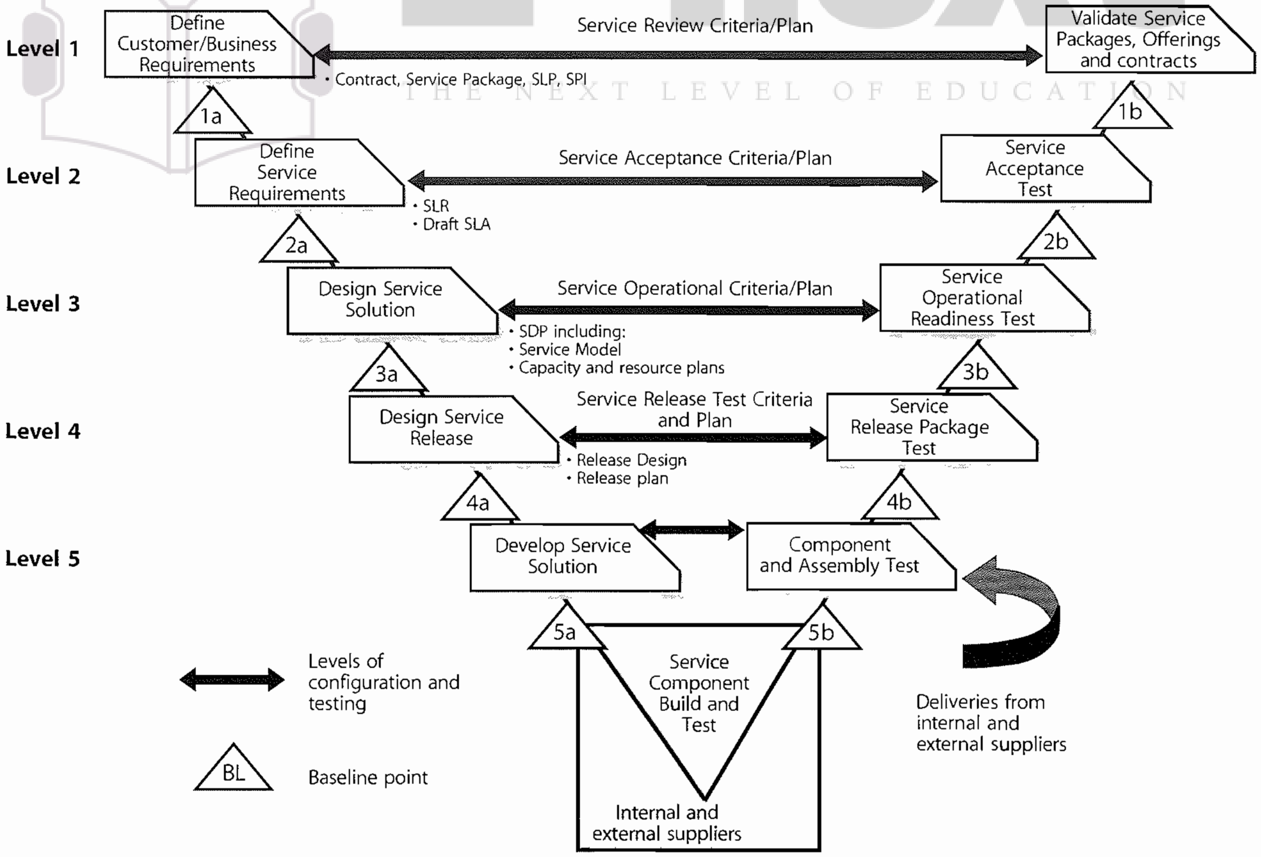
Baselines are added to the CMS as they are developed. Changes to baselines and the release of work products built from the CMS are systematically controlled and monitored via the configuration control, Change Management and configuration auditing functions of SACM. In configuration identification, define and record the rationale for each baseline and associated authorizations required to approve the configuration baseline data.

As a Service progresses through the service lifecycle, each baseline provides progressively greater levels of detail regarding the eventual outputs to be delivered. Furthermore, this hierarchy of baselines enables the final outputs to be traced back to the original requirements.

It needs to be kept in mind that earlier baselines may not be totally up to date with changes that have been made later, e.g. 'course corrections' to requirements documentation may be reflected in the release documentation.

Identification of release unit

'Release unit' describes the portion of the service or infrastructure that is normally released together in accordance with an organization's release policy. The unit may vary, depending on the type(s) or item(s) of software and hardware.



Level 1
Define Customer/Business Requirements
Service Review Criteria/Plan
• Contract, Service Package, SLP, SPI
Validate Service Packages, Offerings and contracts
1a
1b
Level 2
Define Service Requirements
Service Acceptance Criteria/Plan
• SLR
• Draft SLA
Service Acceptance Test
2a
2b
Level 3
Design Service Solution
Service Operational Criteria/Plan
• SDP including:
• Service Model
• Capacity and resource plans
Service Operational Readiness Test
3a
3b
Level 4
Design Service Release
Service Release Test Criteria and Plan
• Release Design
• Release plan
Service Release Package Test
4a
4b
Level 5
Develop Service Solution
Component and Assembly Test
5a
5b
Service Component Build and Test
Internal and external suppliers
Deliveries from internal and external suppliers
Levels of configuration and testing
BL
Baseline point

Figure 4.12 Example of service lifecycle configuration levels and baseline points, represented by the numbered triangles

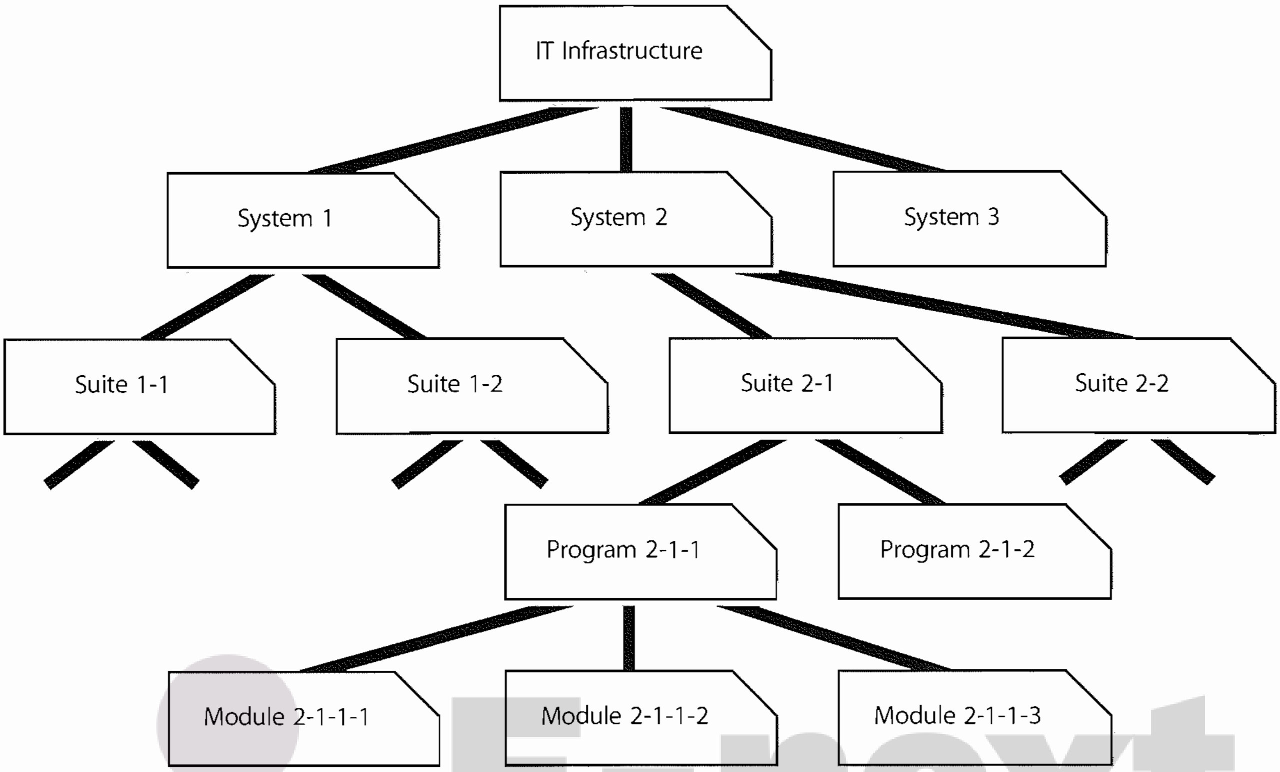


Figure 4.13 Simplified example of an IT infrastructure

Figure 4.13 gives a simplified example showing an IT infrastructure made up of systems, which are in turn made up of suites, comprising programs, which are made up of modules.

Release information is recorded within the CMS, supporting the release and deployment process. Releases are uniquely identified according to a scheme defined in the release policy. The release identification includes a reference to the CI that it represents and a version number that will often have two or three parts. Example release names are:

- Major releases: Payroll_System v.1, v.2, v.3 etc.
- Minor releases: Payroll_System v.1.1, v.1.2, v.1.3 etc.
- Emergency fix releases: Payroll_System v.1.1.1, v.1.1.2, v.1.1.3 etc.

4.3.5.4 Configuration control

Configuration control ensures that there are adequate control mechanisms over CIs while maintaining a record of changes to CIs, versions, location and custodianship/ownership. Without control of the physical or electronic assets and components, the configuration data and information there will be a mismatch with the physical world.

No CI should be added, modified, replaced or removed without an appropriate controlling documentation or procedure being followed. Policies and procedures need to be in place that cover:

- Licence control, to ensure that the correct number of people are using licences and that there is no unlicensed use and no wastage
- Change Management
- Version control of service asset, software and hardware versions, images/builds and releases
- Access control, e.g. to facilities, storage areas and CMS
- Build control, including the use of build specification from the CMS to perform a build
- Promotion, migration of electronic data and information
- Taking a configuration baseline of assets or CIs before performing a release (into system, acceptance test and production) in a manner that can be used for subsequent checking against actual deployment
- Deployment control including distribution
- Installation
- Maintaining the integrity of the DML.

Often there are many procedures that can change a CI. These should be reviewed and aligned with the CI types where possible as standardization prevents errors. During the planning stage it is important to design an effective configuration control model and implement this in a way that staff can easily locate and use the associated training products and procedures.

If many Configuration Management tools are used there is often a control plan for each tool that is aligned with the overall configuration control model.

Control should be passed from the project or supplier to the service provider at the scheduled time with accurate configuration information, documentation and records. A comprehensive checklist covering the service provider information requirements, Supplier information and organizational information required can be made and signed off. Provisions for conducting SACM need to be established in supplier agreements. Methods to ensure that the configuration data is complete and consistent should be established and maintained. Such a method may include baseline on transition, defined audit policies and audit intervals. It is important that the need for this information and control method is established as early as possible during the development lifecycle and incorporated as a deliverable of the new or changed service.

4.3.5.5 Status accounting and reporting

Each asset or CI will have one or more discrete states through which it can progress. The significance of each state should be defined in terms of what use can be made of the asset or CI in that state. There will typically be a range of states relevant to the individual asset or CIs.

A simple example of a lifecycle is:

- Development or draft – denoting that the CI is under development and that no particular reliance should be placed on it
- Approved – meaning that the CI may be used as a basis for further work
- Withdrawn – meaning withdrawn from use, either because the CI is no longer fit for purpose or because there is no further use for it.

The way CIs move from one state to another should be defined, e.g. an application release may be registered, accepted, installed or withdrawn. An example of a lifecycle for a package application release is shown in Figure 4.14. This will include defining the type of review and approval required and the authority level necessary to give that approval. In Figure 4.12 the role that can promote the CI from Accepted to Installed is 'release management'. At each lifecycle status change the CMS should be updated with the reason, date-time stamp and person that did the status change. The planning activities should also establish any attributes that should be updated at each state.

Configuration status accounting and reporting is concerned with ensuring that all configuration data and documentation is recorded as each asset or CI progresses through its lifecycle. It provides the status of the configuration of a service and its environment as the configuration evolves through the service lifecycle.

Status reporting provides the current and historical data concerned with each CI that in turn enables tracking of changes to CIs and their records, i.e. tracking the status as a CI changes from one state to another, e.g. 'development', 'test', 'live' or 'withdrawn'.

The organization should perform configuration status accounting and reporting activities throughout the lifecycle of the service in order to support and enable an efficient Configuration Management process. Typical activities include:

- Maintaining configuration records through the service lifecycle and archiving them according to agreements, relevant legislation or best industry practice, standards, e.g. ISO 9001, Quality Management System
- Managing the recording, retrieval and consolidation of the current configuration status and the status of all preceding configurations to confirm information correctness, timeliness, integrity and security
- Making the status of items under Configuration Management available throughout the lifecycle, e.g. to ensure appropriate access, change, build and release controls are followed, e.g. build specifications
- Recording changes to CIs from receipt to disposal

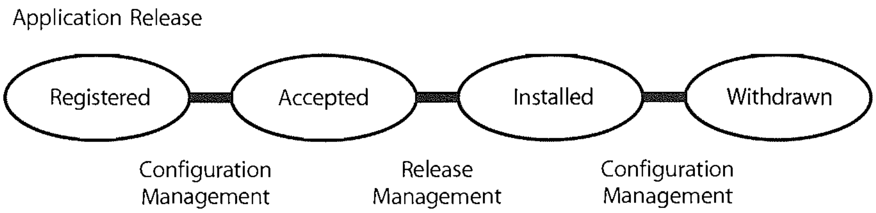


Figure 4.14 Example of asset and configuration item lifecycle

- Ensuring that changes to configuration baselines are properly documented. This can be achieved by consolidating the evolving configuration states of configuration items to form documented baselines at designated times or under defined circumstances.

Records

During the configuration identification and control activities, configuration status records will be created. These records allow for visibility and traceability and for the efficient management of the evolving configuration. They typically include details of:

- Service configuration information (such as identification number, title, effective dates, version, status, change history and its inclusion in any baseline)
- The service or product configuration (such as design or build status)
- The status of release of new configuration information
- Changes implemented and in progress
- Capturing the results from quality assurance tests to update the configuration records.

The evolving service configuration information should be recorded in a manner that identifies the cross-references and interrelationships necessary to provide the required reports.

Service asset and configuration reports

Reports of varying types will be needed for Configuration Management purposes. Such reports may cover individual configuration items, a complete service or the full Service Portfolio. Typical reports include:

- A list of product configuration information included in a specific configuration baseline
- A list of configuration items and their configuration baselines
- Details of the current revision status and change history
- Status reports on changes, waivers and deviations
- Details of the status of delivered and maintained products concerning part and traceability numbers
- Revision status
- Report on unauthorized usage of hardware and software
- Unauthorized CIs detected
- Variations from CMS to physical audit reports.

Status reports of assets for a business unit or software licence holdings are often required by financial management for budgeting, accounting and charging.

4.3.5.6 Verification and audit

The activities include a series of reviews or audits to:

- Ensure there is conformity between the documented baselines (e.g. agreements, interface control documents) and the actual business environment to which they refer
- Verify the physical existence of CIs in the organization or in the DML and spares stores, the functional and operational characteristics of CIs and to check that the records in the CMS match the physical infrastructure
- Check that release and configuration documentation is present before making a release.

Before a major release or change, an audit of a specific configuration may be required to ensure that the customer's environment matches the CMS. Before acceptance into the live environment, new releases, builds, equipment and standards should be verified against the contracted or specified requirements. There should be a test certificate that proves that the functional requirements of a new or updated CI have been verified, or some other relevant document (e.g. RFC).

Plans should be made for regular configuration audits to check that the CMDB and related configuration information is consistent with the physical state of all CIs, and vice versa. Physical configuration audits should be carried out to verify that the 'as-built' configuration of a CI conforms to its 'as-planned' configuration and its associated documents. Interrogation facilities are required to check that the CMDB and the physical state of CIs are consistent.

These audits should verify that correct and authorized versions of CIs exist, and that only such CIs exist, and are in use in the operational environment. From the outset, any ad-hoc tools, test equipment, personal computers and other unregistered items should either be removed or registered through formal Configuration Management. Registration or removal will be via the Change Management process and has to prevent the authorization of non-acceptable CIs or the removal of CIs that may be supporting business processes. Unregistered and unauthorized items that are discovered during configuration audits should be investigated, and corrective action taken to address possible issues with procedures and the behaviour of personnel. All exceptions are logged and reported.

Configuration audits check in addition that change and release records have been properly authorized by Change Management and that implemented changes are as authorized. Configuration audits should be considered at the following times:

- Shortly after changes to the CMS
- Before and after changes to the IT services or infrastructure
- Before a release or installation to ensure that the environment is as expected
- Following recovery from disasters and after a 'return to normal' (this audit should be included in contingency plans)
- At planned intervals
- At random intervals
- In response to the detection of any unauthorized CIs.

Automated audit tools enable regular checks to be made at regular intervals, e.g. weekly. For example, desktop audit tools compare the build of an individual's desktop to the master build that was installed. If exceptions are found, some organizations return the build to its original state.

A rolling programme of configuration audits can help use resources more effectively. The service desk and support groups will check that CIs brought to their attention, e.g. the software that a caller is using, are as recorded in the CMS. Any deviations are reported to Configuration Management for investigation.

If there is a high incidence of unauthorized CIs detected, the frequency of configuration audits should be increased, certainly for those parts of the services or IT infrastructure affected by this problem. Note that unauthorized installations are discouraged when the Configuration Management team is seen to be in control and to carry out regular and frequent audits. If an epidemic of unauthorized CIs is detected, selective or general configuration audits should be initiated to determine the scale of the problem, to put matters right, and to discourage a proliferation of unauthorized CIs. Publicity will help to reduce further occurrences. Service Design and Service Operations staff need to be notified and involved in the investigation of unauthorized CIs.

4.3.6 Triggers, input and output, and inter-process interfaces

Updates to asset and configuration information are triggered by change requests, purchase orders, acquisitions and service requests.

4.3.6.1 Process relationships

By its very nature – as the single virtual repository of configuration data and information for IT Service Management – SACM supports and interfaces with every other process and activity to some degree. Some of the more noteworthy interfaces are:

- Change Management – identifying the impact of proposed changes
- Financial management – capturing key financial information such as cost, depreciation methods, owner and user (for budgeting and cost allocation), maintenance and repair costs
- ITSCM – awareness of assets the business services depend on, control of key spares and software
- Incident/problem/error – providing and maintaining key diagnostic information; maintenance and provision of data to the service desk
- Availability management in detection of points of failure.

The relationship with change and release and deployment is synergistic, with these processes benefiting greatly from a single coordinated planning approach. Configuration control is synonymous with change control – understanding and capturing updates to the infrastructure and services.

4.3.7 Information management

Backup copies of the CMS should be taken regularly and securely stored. It is advisable for one copy to be stored at a remote location for use in the event of a disaster. The frequency of copying and the retention policy will depend on the size and volatility of the IT infrastructure and the CMS. Certain tools may allow selective copying of CI records that are new or have been changed.

The CMS contains information on backup copies of CIs. It will also contain historical records of CIs and CI versions that are archived, and possibly also of deleted CIs or CI versions. The amount of historical information to be retained depends on its usefulness to the organization. The retention policy on historical CI records should be regularly reviewed, and changed if necessary. If the cost to the organization of retaining CI information is greater than the current or potential value, do not retain it, taking note of relevant regulatory and statutory requirements in relation to retention of records.

Typically, the CMS should contain records only for items that are physically available or could be easily created using procedures known to, and under the control of, Configuration Management. When Configuration

Management has been operating for a period of time, regular housekeeping should be carried out to ensure that redundant CI records are systematically archived.

4.3.8 Key performance indicators and metrics

As with all processes the performance of SACM should be monitored, reported on and action taken to improve it.

SACM is the central support process facilitating the exchange of information with other processes and as such has few customer facing measures. However, as an underlying engine to other processes in the lifecycle, SACM must be measured for its contribution to these parts of the lifecycle and the overall KPIs that affect the customer directly.

In order to optimize the cost and performance of the service assets and configurations the following measures are applicable:

- Percentage improvement in maintenance scheduling over the life of an asset (not too much, not too late)
- Degree of alignment between provided maintenance and business support
- Assets identified as the cause of service failures
- Improved speed for incident management to identify faulty CIs and restore service
- Impact of incidents and errors affecting particular CI types, e.g. from particular suppliers or development groups, for use in improving the IT services
- Percentage re-use and redistribution of under-utilized resources and assets
- Degree of alignment of insurance premiums with business needs
- Ratio of used licences against paid for licences (should be close to 100%)
- Average cost per user for licences (i.e. more effective charging options achieved)
- Achieved accuracy in budgets and charges for the assets utilized by each customer or business unit
- Percentage reduction in business impact of outages and incidents caused by poor Asset and Configuration Management
- Improved audit compliance.

Other measures include:

- Increased quality and accuracy of asset and configuration information
- Fewer errors caused by people working with out-of-date information
- Shorter audits as quality asset and configuration information is easily accessible
- Reduction in the use of unauthorized hardware and software, non-standard and variant builds that increase complexity, support costs and risk to the business services
- Reduction in the average time and cost of diagnosing and resolving incidents and problems (by type)
- Improvement in time to identify poor-performing and poor-quality assets
- Occasions when the 'configuration' is not as authorized
- Changes that were not completed successfully or caused errors because of poor impact assessment, incorrect data in the CMS, or poor version control
- Exceptions reported during configuration audits
- Value of IT components detected in use
- Reduction in risks due to early identification of unauthorized change.

4.3.9 Challenges, critical success factors and risks

Challenges to SACM include:

- Persuading technical support staff to adopt a checking in/out policy – this can be perceived as being a hindrance to a fast and responsive support service; if the positives of such a system are not conveyed adequately then staff may be inclined to try and circumvent it; even then, resistance can still occur; placing this as an objective within their annual appraisal is one way to help enforce the policy
- Attracting and justifying funding for SACM, since it is typically out of sight to the customer units empowered with funding control; in practice it is typically funded as an 'invisible' element of Change Management and other ITSM process with more business visibility
- An attitude of 'just collecting data because it is possible to do'; this leads SACM into a data overload which is impossible, or at least disproportionately expensive, to maintain
- Lack of commitment and support from management who do not understand the key role it must play supporting other processes.

Critical success factors include:

- Focusing on establishing valid justification for collecting and maintaining data at the agreed level of detail

- Demonstrating a top-down approach – focused on identifying service CIs and subsequently the CIs that support those services, thereby allowing a rapid and clear demonstration of potential points of failure for any given service
- Setting a justified level of accuracy, i.e. the correlation between the logical model within SACM and the 'real world'
- Making use of enabling technology to automate the CMS practices and enforce SACM policies.

Risks to successful SACM include:

- The temptation to consider it technically focused, rather than service and business focused, since technical competence is essential to its successful delivery
- Degradation of the accuracy of configuration information over time that can cause errors and be difficult and costly to correct
- The CMS becomes out of date due to the movement of hardware assets by non-authorized staff; half-yearly physical audits should be conducted with discrepancies highlighted and investigated; managers should be informed of inconsistencies in their areas.

4.4 RELEASE AND DEPLOYMENT MANAGEMENT

Release and Deployment Management aims to build, test and deliver the capability to provide the services specified by Service Design and that will accomplish the stakeholders' requirements and deliver the intended objectives.

4.4.1 Purpose, goal and objective

The purpose of Release and Deployment Management is to:

- Define and agree release and deployment plans with customers and stakeholders
- Ensure that each release package consists of a set of related assets and service components that are compatible with each other
- Ensure that integrity of a release package and its constituent components is maintained throughout the transition activities and recorded accurately in the CMS
- Ensure that all release and deployment packages can be tracked, installed, tested, verified, and/or uninstalled or backed out if appropriate
- Ensure that organization and stakeholder change is managed during the release and deployment activities (see section 5).
- Record and manage deviations, risks, issues related to the new or changed service and take necessary corrective action
- Ensure that there is knowledge transfer to enable the customers and users to optimize their use of the service to support their business activities
- Ensure that skills and knowledge are transferred to operations and support staff to enable them to effectively and efficiently deliver, support and maintain the service according to required warranties and service levels.

The goal of Release and Deployment Management is to deploy releases into production and establish effective use of the service in order to deliver value to the customer and be able to handover to service operations.

The objective of Release and Deployment Management is to ensure that:

- There are clear and comprehensive release and deployment plans that enable the customer and business change projects to align their activities with these plans
- A release package can be built, installed, tested and deployed efficiently to a deployment group or target environment successfully and on schedule
- A new or changed service and its enabling systems, technology and organization are capable of delivering the agreed service requirements, i.e. utilities, warranties and service levels
- There is minimal unpredicted impact on the production services, operations and support organization
- Customers, users and Service Management staff are satisfied with the Service Transition practices and outputs, e.g. user documentation and training.

4.4.2 Scope

The scope of Release and Deployment Management includes the processes, systems and functions to package, build, test and deploy a release into production and establish the service specified in the Service Design package before final handover to service operations.

4.4.3 Value to business

Effective Release and Deployment Management enables the service provider to add value to the business by:

- Delivering change, faster and at optimum cost and minimized risk

- Assuring that customers and users can use the new or changed service in a way that supports the business goals
- Improving consistency in implementation approach across the business change, service teams, suppliers and customers
- Contributing to meeting auditable requirements for traceability through Service Transition.

Well-planned and implemented release and deployment will make a significant difference to an organization's service costs. A poorly designed release or deployment will, at best, force IT personnel to spend significant amounts of time troubleshooting problems and managing complexity. At worst, it can cripple the environment and degrade the live services.

4.4.4 Policies, principles and basic concepts

4.4.4.1 Release unit and identification

A 'release unit' describes the portion of a service or IT infrastructure that is normally released together according to the organization's release policy. The unit may vary, depending on the type(s) or item(s) of service asset or service component such as software and hardware. Figure 4.15 gives a simplified example showing an IT service made up of systems and service assets, which are in turn made up of service components.

The general aim is to decide the most appropriate release-unit level for each service asset or component. An organization may, for example, decide that the release unit for business critical applications is the complete application in order to ensure that testing is comprehensive. The same organization may decide that a more appropriate release unit for a website is at the page level.

The following factors should be taken into account when deciding the appropriate level for release units:

- The ease and amount of change necessary to release and deploy a release unit
- The amount of resources and time needed to build, test, distribute and implement a release unit
- The complexity of interfaces between the proposed unit and the rest of the services and IT infrastructure
- The storage available in the build, test, distribution and live environments.

Releases should be uniquely identified according to a scheme defined in the release policy as discussed in section 4.1.4.2. The release identification should include a reference to the CIs that it represents and a version number that will often have two or three parts, e.g. emergency fix releases: Payroll_System v.1.1.1, v.1.1.2, v.1.1.3.

4.4.4.2 Release design options and considerations

Service Design will define the approach to transitioning from the current service to the new or changed service or service offering. The SDP defines the service and solution design components to be transitioned to deliver the required service package(s) and service level package(s).

Common options for release and deployment that are considered in Service Design are discussed below. The selected option will have a significant impact on the release and deployment resources as well as the business outcomes. It is important to understand the patterns of

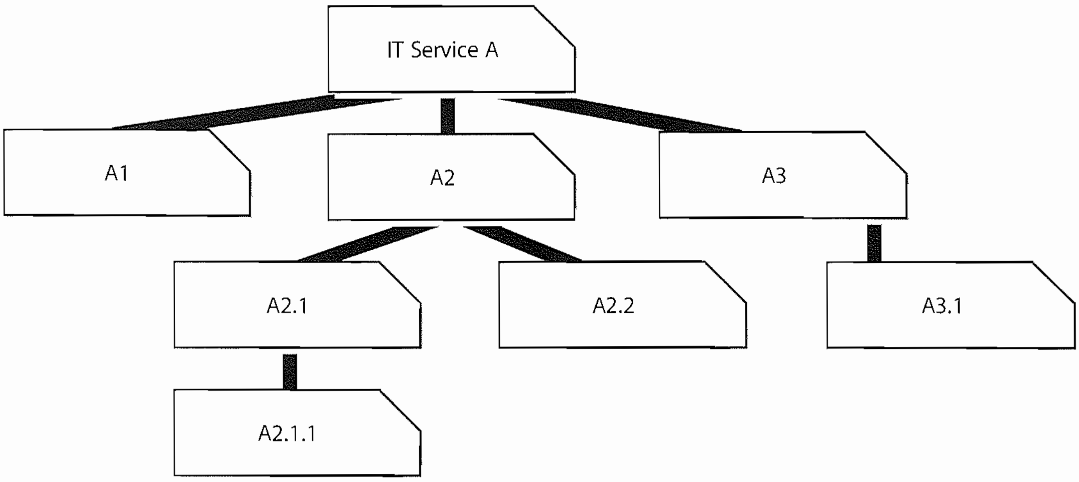


Figure 4.15 Simplified example of release units for an IT service

business activity (PBA) and user profiles when planning and designing the releases.

'Big bang' vs phased

Options for deploying new releases to multiple locations are illustrated in Figure 4.16 and described below:

- 'Big bang' option – the new or changed service is deployed to all user areas in one operation. This will often be used when introducing an application change and consistency of service across the organization is considered important.
- Phased approach – the service is deployed to a part of the user base initially, and then this operation is repeated for subsequent parts of the user base via a scheduled rollout plan. This will be the case in many scenarios such as in retail organizations for new services being introduced into the stores' environment in manageable phases.

Figure 4.16 also illustrates a possible sequence of events over time as follows:

- There is an initial launch of the 'Release 1' of the system to three workstations (1–3).
- Two further workstations (4+5) are then added at the same time.
- 'Release 2' of the system is then rolled out in a 'big bang' approach to all workstations (1–5) at once.
- Two further workstations (6+7) are then added, in another step.
- There is a phased implementation of the upgrade to 'Release 3' of the system, initially upgrading only three workstations (1–3) and then the remaining four (4–7).
- A further workstation (8) is then added to the system.

Variations of the phased approach include:

- Portions of the service are delivered to the live environment in phases, but all end users are affected simultaneously (e.g. incremental changes to a shared application).
- Each release is deployed gradually across the total population of end users (e.g. one geographical location at a time).
- Different types of service element are deployed in separate phases, e.g. hardware changes are first, followed by user training and then by the new or changed software.
- A combination of all of these approaches is usually adopted, and the plans may deliberately allow for variations in the light of actual deployment experience.

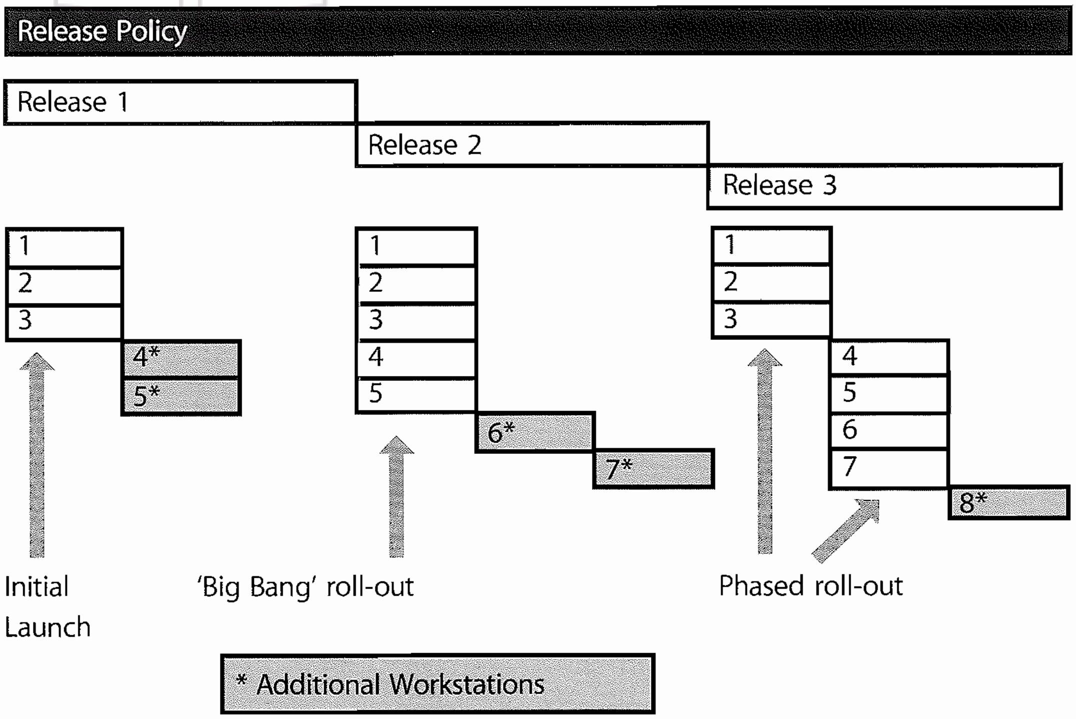


Figure 4.16 Options for 'big bang' and phased rollout

In the type of phased implementation illustrated above, it is only possible to employ this approach if the service has been designed to allow new and old versions to coexist. If this is not possible then the only alternative is to upgrade all affected parts together in a 'big bang' implementation. For elements such as documentation, for skilled staff this is rarely a problem; for many instances of hardware and software it is possible. For other transitions, such as those involving major network changes, it can be virtually impossible to achieve.

Figure 4.17 illustrates phased deployment to a number of different geographical locations. It assumes that new versions will work alongside the previous one. The example used assumes that new functionality is implemented first in the head office of the organization, then in a pilot branch and finally in the remaining branches. If there are a very large number of locations to deal with, it may still take a long time to implement the initial system or upgrades in all branches, thus increasing the likelihood of needing to support even more versions of the system in the live environment concurrently.

Push and pull

A push approach is used where the service component is deployed from the centre and pushed out to the target locations. In terms of service deployment, delivering updated service components to all users – either in big-bang or phased form – constitutes 'push', since the new or changed service is delivered into the users' environment at a time not of their choosing.

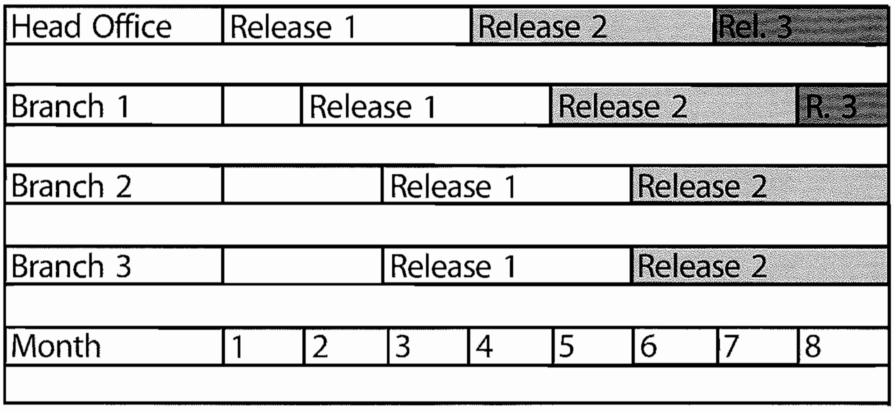
A pull approach is used for software releases where the software is made available in a central location but users are free to pull the software down to their own location at a time of their choosing or when a user workstation restarts. The use of 'pull' updating a release over the internet has made this concept significantly more pervasive. A good example is virus signature updates, which are typically pulled down to update PCs and servers when it best suits the customer; however at times of extreme virus risk this may be overridden by a release that is pushed to all known users.

In order to deploy via 'push' approach, the data on all user locations must be available. Pull approaches do not rest so heavily on accurate configuration data and they can trigger an update to user records. This may be through new users appearing and requesting downloads or expected users not doing so, triggering investigation into their continued existence. As some users will never 'pull' a release it may be appropriate to allow a 'pull' within a specified time limit and if this is exceeded a push will be forced, e.g. for an anti-virus update.

Automation vs manual

Whether by automation or other means, the mechanisms to release and deploy the correctly configured service components should be established in the release design phase and tested in the build and test stages.

Automation will help to ensure repeatability and consistency. The time required to provide a well-designed and efficient automated mechanism may not always be available or viable. If a manual mechanism is used it is important to monitor and measure the impact of many repeated manual activities as they are likely to be inefficient and error-prone. Too many manual activities will slow down the release team and create resource or capacity issues that affect the service levels.



Head Office	Release 1			Release 2			Rel. 3	
Branch 1		Release 1			Release 2			R. 3
Branch 2			Release 1			Release 2		
Branch 3			Release 1			Release 2		
Month	1	2	3	4	5	6	7	8

A phased roll-out across several geographical locations

Figure 4.17 Phased deployment across geographical locations

Many of the release and deployment activities are capable of a degree of automation. For example:

- Discovery tools aid release planning.
- Discovery and installation software can check whether the required prerequisites and co-requisites are in place before installation of new or changed software components.
- Automated builds can significantly reduce build and recovery times that in turn can resolve scheduling conflicts and delays.
- Automated configuration baseline procedures save time and reduce errors in capturing the status of configurations and releases during build, test and deployment.
- Automatic comparisons of the actual 'live' configuration with the expected configuration or CMS help to identify issues at the earliest opportunity that could cause incidents and delays during deployment.
- Automated processes to load and update data to the CMS help to ensure the records are accurate and complete.
- Installation procedures automatically update user and licence information in the CMS.

Designing release and release packages

Figure 4.18 provides an example of how the architectural elements of a service may be changed from the current baseline to the new baseline with releases at each level. The architecture will be different in some organizations but is provided in this section to give a context for release and deployment activities. The release and deployment teams need to understand the relevant architecture in order to be able to plan, package, build and test a release to support the new or changed service. This helps to prioritize the release and deployment activities and manage dependencies, e.g. the technology infrastructure needs to be ready with operations staff ready to support it with new or changed procedures before an application is installed.

Figure 4.18 also shows how the service architectural elements depend on the Service Portfolio that defines the service offerings and service packages. Dependent services will need to be built and tested in Service Transition. For example an IT financial service may be dependent on several internal support services and an external service. For more details about the structure of services, see the Service Strategy and Service Design publications.

There are normally dependencies between particular versions of service components required for the service to operate. For example a new version of an application may require an upgrade to the operating system and one or other of these two changes could require a hardware change, e.g. a faster processor or more memory. In some cases, the release package may consist of a set of documentation and procedures. These could be deployed via a manual update or through an automatic publishing mechanism, e.g. to the SKMS/website.

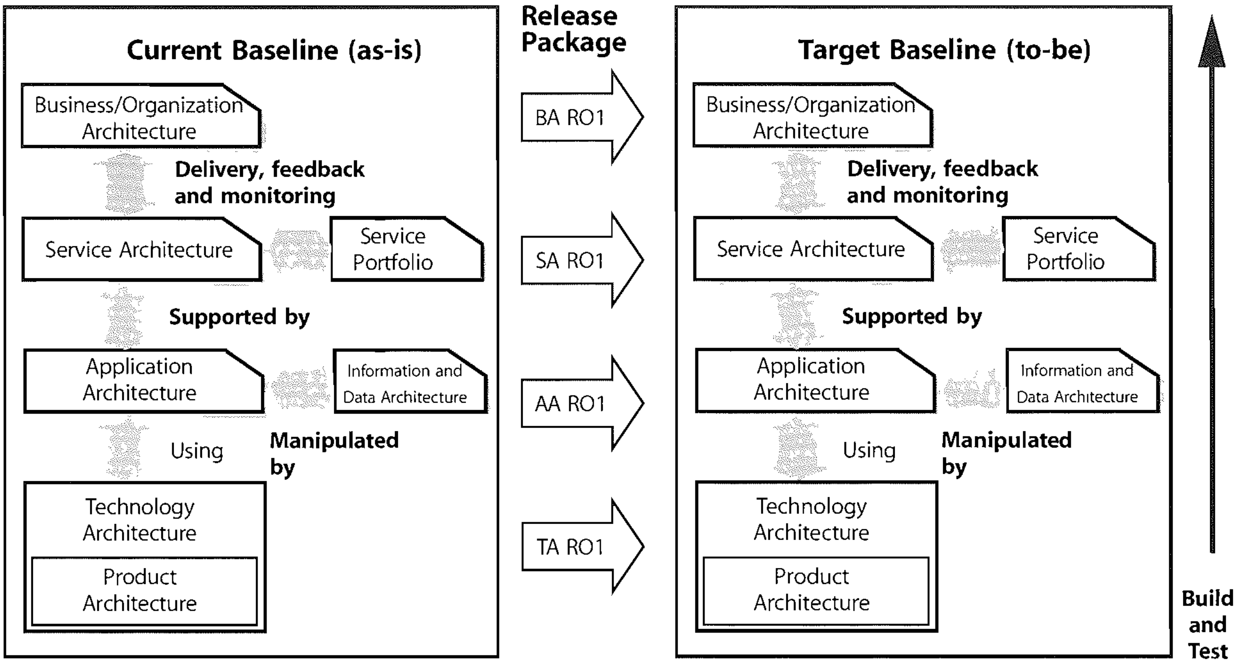


Figure 4.18 Architecture elements to be built and tested

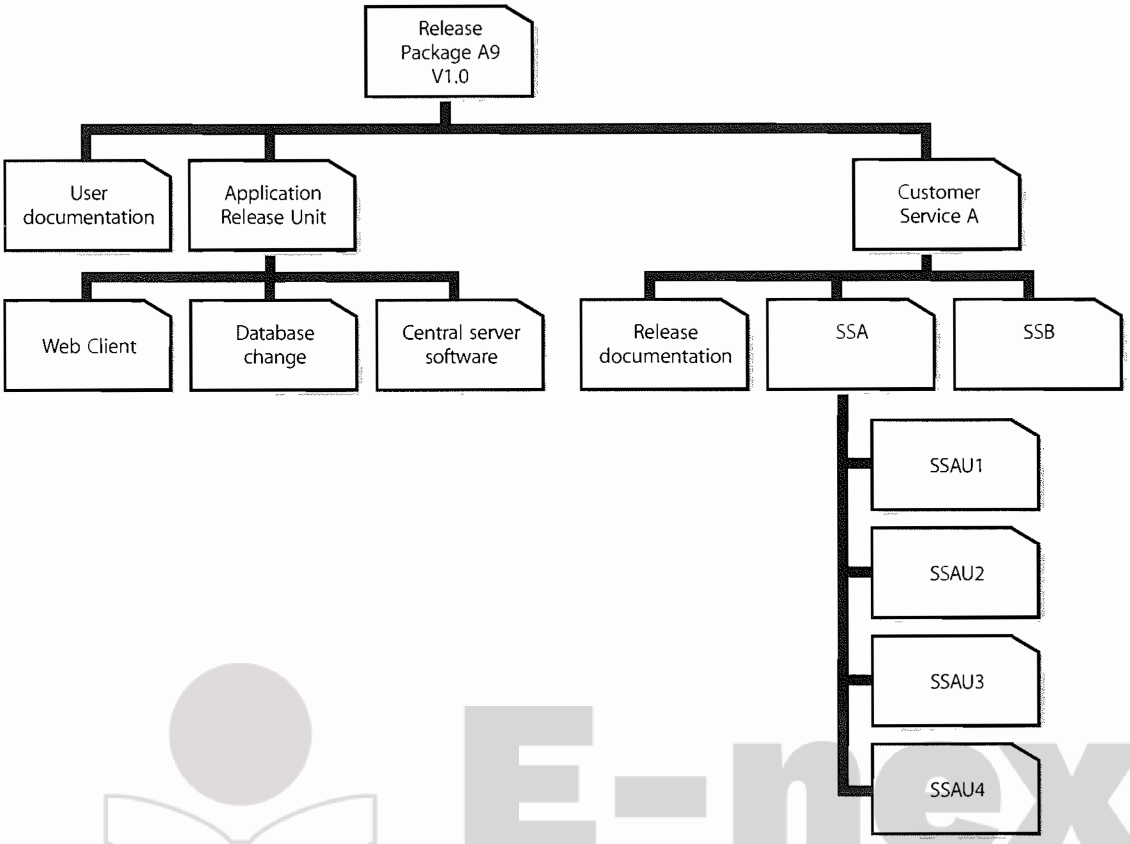


Figure 4.19 Example of a release package

A release package may be a single release unit or a structured set of release units such as the one shown in Figure 4.19.

The example in Figure 4.19 shows an application with its user documentation and a release unit for each technology platform. On the right there is the customer service asset that is supported by two supporting services – SSA for the infrastructure service and SSB for the application service. These release units will contain information about the service, its utilities and warranties and release documentation. Often there will be different ways of designing a release package and consideration needs to be given to establishing the most appropriate method for the identifiable circumstances, stakeholders and possibilities.

Where possible, release packages should be designed so that some release units can be removed if they cause issues in testing.

Valuable release windows

A UK government department is especially well placed to make full use of all available release windows. They work in a secure financial, low risk environment, with carefully planned changes scheduled well in advance and allocated to pre-arranged release windows, which are scheduled several months apart. Because of their careful and longer term planning, when a change proves unsuitable for release, i.e. tests are failed, alternative, quality-assured changes are usually available – prepared and tested but lower in business priority and so targeted at later releases. These can be accelerated to make use of the unexpected vacancy created by the test failure. The test and build process also allows elements of later scheduled releases to be slotted in for release, or successful components of the failed release to be implemented, even though the full products is not ready. This allows subsequent fuller release to be a 'smaller' product, therefore allowing further additional changes to be scheduled alongside them in later release windows.

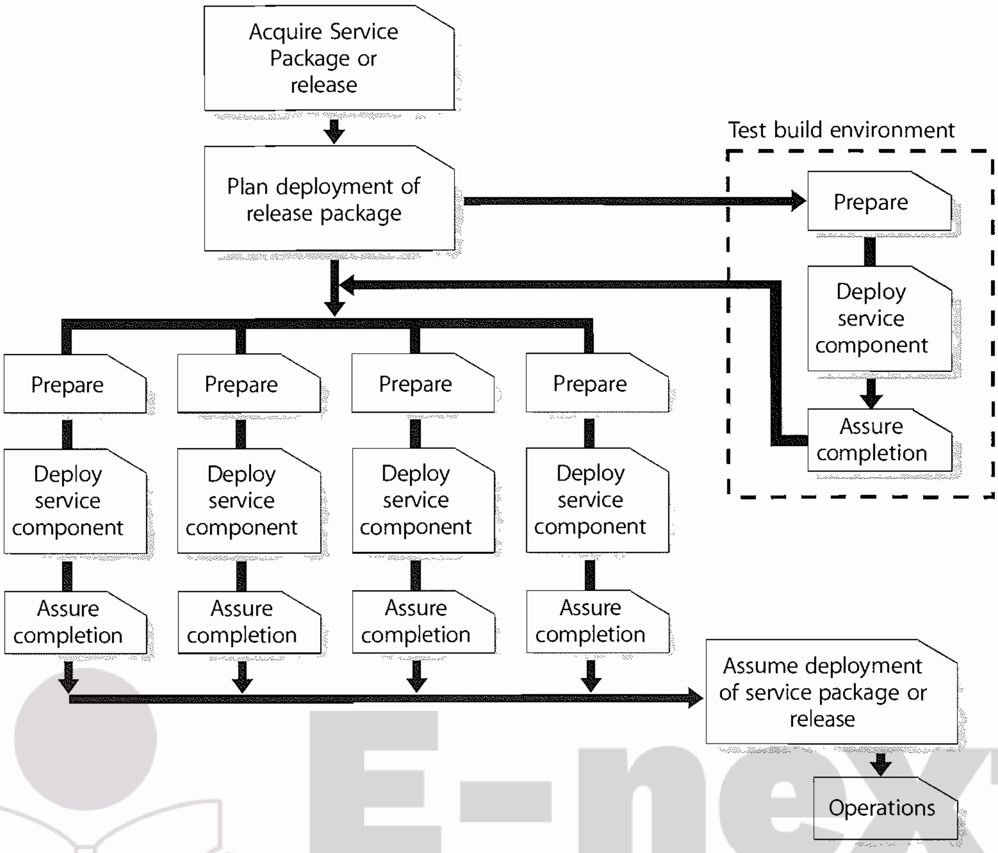


Figure 4.20 Coordinating the deployment of service components

Any significant new or changed service or service offering will require the deployment stage to consider the full range of elements comprising that service – infrastructure, hardware, software, applications, documentation, knowledge etc. Effectively this means the deployment will contain sub-deployments for elements comprising the service, as illustrated in Figure 4.20. The combination, relationship and interdependencies of these components will require careful and considered planning. Significant deployments will be complex projects in their own right.

To understand the deployment options a high level assessment of the deployment units, locations and environments may be required, for example:

- Assessment baseline – this is a snapshot of the relevant environment, services and infrastructure, including 'softer' elements such as skills level and attitudes where applicable, should be taken as a first step.
- Identify the components – this may include deciding the best way to break down a major deployment into components. Often there will be different ways of achieving this breakdown and consideration needs to be given to establishing the most appropriate method for all the identifiable circumstances, stakeholders and possibilities.
- Determine the appropriate deployment approach for each.

4.4.4.3 Release and deployment models

A service may be deployed into the production environment in a number of ways. Service Design will select the most suitable release and deployment models that include the approach, mechanisms, processes, procedures and resources required to build and deploy the release on time and within budget.

The release methods during the early build and test stages may differ significantly from live operations so plan ahead to ensure that appropriate release methods are adopted at the right time.

Release and deployment models define:

- Release structure – the overall structure for building a release package and the target environments
- The exit and entry criteria including mandatory and optional deliverables and documentation for each stage

- Controlled environments required to build and test the release for each release level; there will be multiple logical and physical environments through the Service Transition stage mapped to different physical environments available to the transition team
- The roles and responsibilities for each configuration item at each release level
- The release promotion and configuration baseline model
- Template release and deployment schedules
- Supporting systems, tools and procedures for documenting and tracking all release and deployment activities
- The handover activities and responsibilities for executing the handover and acceptance for each stage of release and deployment.

Considerations in designing the release and deployment model include activities to:

- Verify that a release complies with the SDP, architecture and related standards
- Ensure the integrity of hardware and software is protected during installation, handling, packaging and delivery
- Use standard release and deployment procedures and tools
- Automate the delivery, distribution, installation, build and configuration audit procedures where appropriate to reduce costly manual steps
- Manage and deploy/re-deploy/remove/retire software licences
- Package and build the release package so that it can be backed out or remediated if required
- Use Configuration Management procedures, the CMS and DML to manage and control components during the build and deployment activities, e.g. to verify the prerequisites, co-requisites and post-installation requests
- Document the release and deployment steps
- Document the deployment group or target environment that will receive the release
- Issue service notifications.

4.4.5 Process activities, methods and techniques

4.4.5.1 Planning

Release and deployment plans

Plans for release and deployment will be linked into the overall Service Transition plan and adopt the selected release and deployment model. The approach is to derive a sound set of guidelines for the release into production and subsequent deployment that can be scaled from small organizations to large multinationals. Although smaller organizations will have less complex environments, the disciplines detailed here are still relevant. Even within a single organization, the release and deployment plans need to be scalable since the extent of their scale of impact on the organization will vary, perhaps from impacting only one small specialist team in one location through to multinational impact on all users when introducing new desktop equipment and services, or transferring services to different suppliers.

Release and deployment plans should be authorized through Change Management. They should define the:

- Scope and content of the release
- Risk assessment and risk profile for the release
- Organizations and stakeholders affected by the release
- Stakeholders that approved the change request for the release and/or deployment
- Team responsible for the release
- Approach to working with stakeholders and deployment groups to determine the:
  - Delivery and deployment strategy
  - Resources for the release and deployment
  - Amount of change that can be absorbed.

Pass/fail criteria

Service Transition is responsible for planning the pass/fail situations. At a minimum these should be defined for each authorization point through the release and deployment stage. It is important to publish these criteria to relevant stakeholders well in advance to set expectations correctly. An example of a pass situation before build and test is:

- All tests are completed successfully; the evaluation report and RFC for build and test are signed off.

Examples of fail situations include:

- Insufficient resources to pass to the next stage. For example, an automated build is not possible and so the resource requirement becomes error-prone, too onerous and expensive; testing identifies that there will not be enough money to deliver the proposed design in the operations phase.
- Service Operation does not have capabilities to offer particular service attributes.
- Service Design does not conform to the service operation standards for technologies, protocols, regulations, etc.

- The service cannot be delivered within the boundaries of the design constraints.
- Service acceptance criteria are not met.
- Mandatory documents are not signed off.
- SKMS and CMS are not updated, perhaps due to a process that is manually intensive.
- The incidents, problems and risks are higher than predicted, e.g. by over 5%.

Build and test prior to production

Build and test planning establishes the approach to building, testing and maintaining the controlled environments prior to production. The activities include:

- Developing build plans from the SDP, design specifications and environment configuration requirements
- Establishing the logistics, lead times and build times to set up the environments
- Testing the build and related procedures
- Scheduling the build and test activities
- Assigning resources, roles and responsibilities to perform key activities, e.g.:
  - Security procedures and checks
  - Technical support
  - Preparing build and test environments
  - Managing test databases and test data
  - Software asset and licence management
  - Configuration Management – configuration audit, build and baseline management
- Defining and agreeing the build exit and entry criteria.

Figure 4.21 provides an example of a model that can be used to represent the different configuration levels to be built and tested to deliver a service capability. The left-hand side represents the specification of the service requirements down to the detailed Service Design. The right-hand side focuses on the validation and test activities that are performed against the specifications defined on the left-hand side. At each stage on the left-hand side, there is direct involvement by the equivalent party on the right-hand side. It shows that service validation and acceptance test planning should start with the definition of the service requirements. For example, customers who sign off the agreed service requirements will also sign off the service Acceptance Criteria and test plan.

The V-model approach is traditionally associated with the waterfall lifecycle, but is, in fact, just as applicable to other lifecycles, including iterative lifecycles, such as prototyping, RAD approaches. Within each cycle of the iterative development, the V-model concepts of establishing acceptance requirements against the requirements and design can apply, with each iterative design being considered for the degree of integrity and competence that would justify release to the customer for trial and assessment.

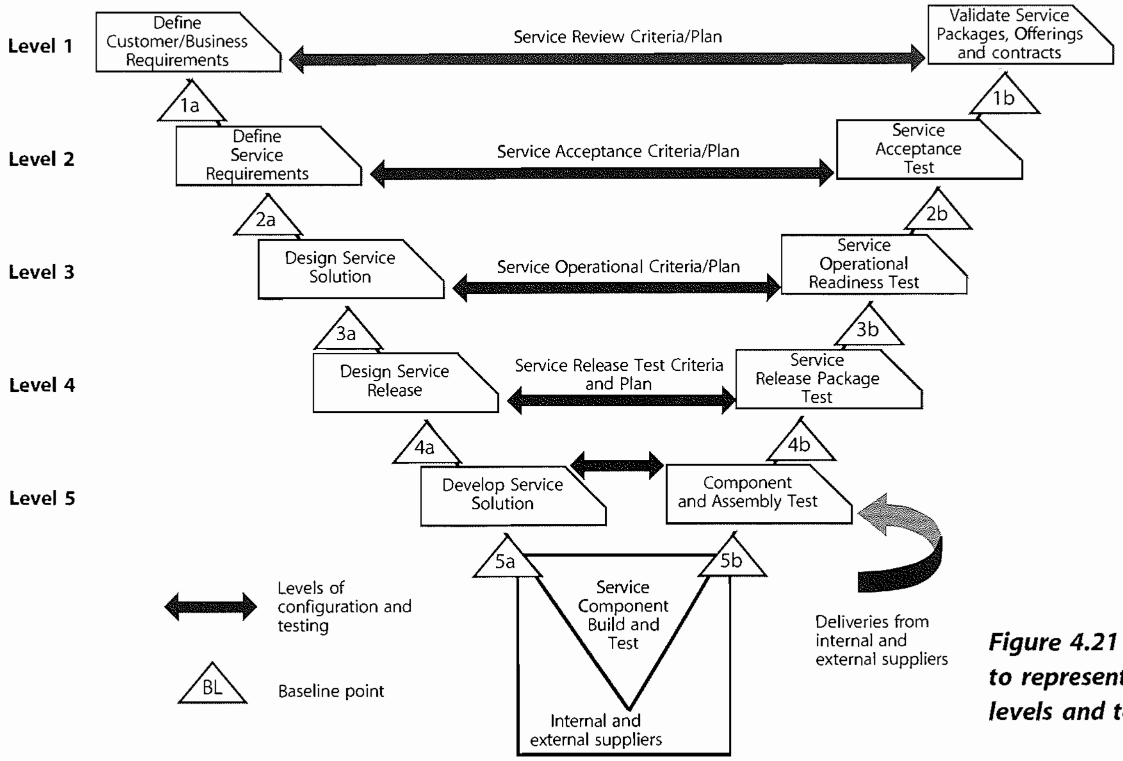

Level 1
Define Customer/Business Requirements
Service Review Criteria/Plan
Validate Service Packages, Offerings and contracts
1a
1b
Level 2
Define Service Requirements
Service Acceptance Criteria/Plan
Service Acceptance Test
2a
2b
Level 3
Design Service Solution
Service Operational Criteria/Plan
Service Operational Readiness Test
3a
3b
Level 4
Design Service Release
Service Release Test Criteria and Plan
Service Release Package Test
4a
4b
Level 5
Develop Service Solution
Component and Assembly Test
5a
5b
Service Component Build and Test
Internal and external suppliers
Deliveries from internal and external suppliers
Levels of configuration and testing
BL
Baseline point

Figure 4.21 Service V-model to represent configuration levels and testing

Table 4.8 Levels of configuration for build and testing

Level	Requirements and design	Build/deliverable	Validation and testing
Level 1 Customer/business needs	Structured definition of contract requirements	Customer contract (based on Service Portfolio, SLP)	Service test and evaluation Determines whether a service can enable the users and customers to use the service to support their business needs (is fit for purpose and fit for use).
Level 2 Service requirements	Service requirement specifications and SAC, traceable back to the contract requirements	Service capability and resources to deliver against the SLA and service requirements	Service test Test the Service Acceptance Criteria are met. Includes validation of service performance against the service level requirements and SLA in pilots, deployment and early life support.
Level 3 Service solution	SDP, Service model, service environments	Solution/system required to deliver the service capability; includes the Service Management and Service Operations systems and capabilities	Service operational readiness test To evaluate the integration and operation of the service capability and resources. It verifies that the target deployment organization and people are prepared to deploy and operate the new or changed service in the live environment, e.g. deployment team, Service Operations, customers, users and other stakeholders. Tests include scenario-based testing such as simulation and service rehearsal.
Level 4 Service release		Release package	Service release test A test that the service components can be integrated correctly and that the release can be installed, built and tested in the target environments. Service release testing includes non-functional testing that can be performed at this level.
Level 5 Component and assemblies	Component and assembly test specification	Component or assembly of components	Component and assembly test Test that a service component or assembly of components matches its detailed specification. Components or assemblies are tested in isolation, with a view to their delivering as specified, in terms of inputs generating expected outputs. Evidence of component quality or testing earlier in the chain may be obtained for test evidence, from both internal and external suppliers.

Further details on validation, testing and service evaluation are provided in sections 4.5 and 4.6. The test strategy defines the overall approach to validation and testing. It includes the organization of validation and testing activities and resources and can apply to the whole organization, a set of services or an individual service.

Typical levels of configuration for build and testing are shown in Table 4.8.

Various controlled environments will need to be built or made available for the different types and levels of testing as well as to support other transition activities such as training. Existing deployment processes and procedures can be used to build the controlled test environments. The environments will need to be secure to ensure there is no unauthorized access and that any segregation of duty requirements are met. The types of environments, both logical and physical, required during release and deployment include:

- Build environments – used to compile or assemble the release package or service assets
- Unit test environment – used for verifying the functionality, performance, recovery and usability

characteristics of an individual service component, e.g. online procedure

- Assembly test environment – used for verifying the functionality, performance, recovery and usability characteristics of an assembly of service components
- Integration environment – for building and integrating service components
- System test environment – used for testing all aspects of the integrated service architecture, including the application and technical infrastructure; substantial user acceptance testing is executed in this environment
- Service release test environment – used to install, build and test a release package in a controlled environment; this is often combined with the system test environment
- Service Operations readiness test environment – for testing the service and service unit capabilities before promotion into live; may include the Service Management acceptance test, some operational acceptance tests and user acceptance tests of the end-to-end service
- Business simulation environments
- Service Management simulation environments
- Training environments – sometimes this may include an established test database that can be used as a safe and realistic training environment
- Pilot environments, including conference room pilots
- Backup and recovery environments, e.g. disaster recovery.

Planning pilots

Pilots are useful for testing the service with a small part of the user base before rolling it out to the whole service community. It is important to determining the appropriate scope of a pilot (how much of the service is to be included in the pilot, size of department or user base). This is a key step in establishing the pilot effort. If the scope is too small then insufficient functionality and implementation variations will be trialled and the likelihood of significant errors not being discovered until full rollout is higher. If the scope is too large it will not deliver the speed and flexibility that deliver the benefits, but will effectively be a first rollout.

A pilot can be used to establish the viability of most, if not all, aspects of the service. But this will only happen if all stakeholders are actively involved in the pilot and use the service as it would be done in a full rollout.

The pilot should include steps to collect feedback on the effectiveness of the deployment plan. This can include:

- Surveying views and satisfaction from:
  - End users
  - Customers
  - Suppliers
  - Service desk and other support staff
- Network management
- Data and Knowledge Management – statistics on use and effectiveness
- Analysing statistics from service desk calls, suppliers, capacity and availability.

Commitment to support the pilot is required from all involved parties. Obtaining that commitment can be a challenge since pilots typically will represent additional work for those users involved over and above their day jobs. Collaboration from suppliers and support staff (who may have to be supporting two versions of a service in parallel, or deliver a small separate unit dedicated to supporting the pilot) must also be obtained.

Planning should accommodate rolling back a pilot before the full rollout of an authorized new service. New services tend to be piloted with test equipment and this needs to be rolled back to its original state. In addition, users who were part of the pilot should be working with the same components of a service as other users after the full rollout, not the setup put in place for the pilot. This simplifies, day-to-day operations in IT Service Management.

Although a pilot is often thought of as one trial in the production environment before rolling a service out across the full customer and user environment, there may be justification for a range of pilots, e.g. a pilot for deployment to each geographical region. Many considerations are relevant, with the best solution for a given circumstance being a balance between benefit and cost. Factors include:

- **Speed and cost** – A single pilot will be cheaper and faster than multiple pilots, and is the obvious choice for a homogeneous organization where a single pilot will encounter (almost) all eventualities and so provide a high degree of confidence that a successful pilot would be followed by a successful roll-out across the wider organization.
- **Diverse organization** – In an organization with a range of circumstances across the user base, or with multiple operating environments, a matching range of pilots may be sensible, with a trial in each of the areas. These can be managed in parallel, with simultaneous trialling in each environment, which reduces elapsed time but increases management overhead and complexity. Alternatively, by running the

pilots serially, lessons learned in one environment may be usefully applied to the subsequent ones, since even in diverse organization there is likely to be significant common ground, e.g. within the actual service components. Examples of significant diversity include:

  - Different training methods needed for different groups
  - Technology
  - Language or culture
  - Network capability.

- **Trialling options** – Where alternative solutions are possible for a major rollout, it may be worth trying each of the options in a separate pilot (preferably in closely matched areas to make comparisons meaningful). Armed with the results from each pilot, a decision as to the approach for the main rollout can be taken based on solid empirical evidence.
- **Political considerations** – Internal or external political issues may mean that a specific group or groups needs to be involved – or not involved – in a pilot for a new or changed service.

Example of need for multiple piloting

A government organization delivers desktop IT services to all their staff – in corporate headquarters (HQ) and in locations throughout the world. When new or significant changes are to be rolled out, typically three parallel pilots are carried out to test the three levels of communication and support technology they have identified:

- Those in HQ on direct network connection and with local dedicated support staff
- Those in larger locations with reliable high-speed connection and semi-specialized local IT administrators
- Those in smaller locations with unreliable communications and no trained local support.

Experience has shown that the three groups have different implementation and support issues and that the pilots in all three types of customer are worth the extra costs and complications.

Planning release packaging and build

Planning the release packaging and build activities includes the activities to develop the mechanisms, plans or procedures for the following:

- Verifying the entry/exit criteria
- Managing stakeholder change and communications by:
  - Obtaining and maintaining the list of contacts and their details
  - Communicating the proposed changes, the expected benefits and how the change affects the organization and staff
- Training people and transferring knowledge
- Establishing the Services and service assets, e.g. agreements and contracts are in place
- Agreeing schedules:
  - Agreeing the delivery schedules and handling any changes/delays
  - Finalising the logistics and delivery procedures and checklists
  - Scheduling and allocating controlled transition environments, facilities and tools for: i) acquisition of service assets and components, and ii) release packaging, building and testing
- Developing procedures and mechanisms using available Configuration Management, release, content/electronic publishing and other tools to:
  - Build, copy, promote, distribute, audit, install and activate a release
  - Manage software licences, digital rights and Intellectual Property Rights (IPR)
- Converting systems and users from the current applications and technology to the new or changed service, e.g. migrate or reformat application data and information
- Developing the Service Management capability and resources for:
  - Conducting site surveys
  - Updating service information, e.g. service catalogue, release documentation
  - Building and preparing the management systems and other operational systems, e.g. systems and event management, measurement systems
  - Operating and handling the predicted capacity required for support
  - Operating the controlled environments including procedures to scale up capacity if required
  - Documenting and providing the information to be created and/or updated during transition, e.g. remediation plans to be issued and published
  - Installing the new or changed service ready for activation
  - Transferring/transitioning a service or service team or organization
  - Decommissioning and/or disposing of service assets and components
  - Retiring services

- Assessing the readiness of a target deployment group (customers, users and Service Operations staff) to take a release
- Defining and agreeing the exit criteria.

Deployment planning

There are many planning considerations that need to be considered. Planners should be able to answer the questions included in Table 4.9.

Logistics and delivery planning

Once the overall deployment approach is understood, develop the logistics and delivery plans. These plans deal with aspects such as:

- How and when release units and service components will be delivered
- What the typical lead times are; what happens if there is a delay
- How to track progress of the delivery and obtain confirmation of delivery
- Availability of secure storage where required
- Managing customs and other implications of international distribution.

As well as the delivery aspects, there are typically consequential logistics to be dealt with, e.g. decommissioning and disposing of redundant items, including software and licences, hardware, skills, computer and staff accommodation, support contracts (utility supply, maintenance, cleaners etc.). There may also be a need for temporary equipment (e.g. swing equipment) or throwaway software that is required for the transition.

If the transition plans call for any parallel running of services or equipment, this is particularly taxing from a logistics perspective, since double facilities are likely to be required for a short time.

Once the logistics and delivery plans have been determined, they need to be communicated to all stakeholders, including formal notification to those consulted in deriving the plan.

Delivery is not sufficient; successful logistics requires that the components arrive and perform as required. Therefore deployment planning for all despatched items – hardware,

Table 4.9 Questions to be answered when planning deployment

Deployment question	Examples
What needs to be deployed?	Do you have a good understanding of the service and release that is being deployed? What are the components that make up the release package? What are the business drivers for the deployment? Is it required to meet a critical business need?
Who are the users?	Which users are affected by the deployment? What language do they use? Do they need any special training?
Are there location dependencies?	Are there any holidays, shut-downs or other interruptions to normal business at this location? What level of detail needs to be recorded, e.g. building, floor, room?
Where are the users?	Are all the users and systems local to the deployment, or are some remote, and how will this affect the logistics?
Who else needs to be prepared well in advance?	Do the service desk and support staff need training? Are there any access issues to be solved – security or physical?
When does the deployment need to be completed?	Does the deployment need to be completed by a certain date and time or can it be completed by following a flexible schedule?
Why is the deployment happening?	Is the deployment needed to fix a problem or is it required for some new functionality that has been requested, and do the users understand what is coming?
What are the critical success factors and exit criteria?	How will you know that the deployment has been successful? Who will authorize the deployment? How will you know when the deployment is finished?
What is the current capability of the service provider?	What are the current services, processes and Service Management capability – capacity, financial aspects, current systems and infrastructure?

software, documentation, and training – will address how components are tracked and documented on delivery. This should include:

- Checking against a definitive list of required service assets and components' unique IDs and versions
- A delivery note detailing the components to be delivered, including unique IDs, versions and quantities
- What there should be (contents list to check against)
- What needs to be there to meet it, in terms of equipment, prerequisites and co-requisites
- How to ensure it is correct/working – what tools, parameters, feedback mechanisms, Acceptance Criteria need to be applied?
- Metrics for monitoring and determining success of the release deployment effort.

Financial/commercial planning

Financial and commercial aspects will need to be specifically checked before the deployment and activities added to the deployment plans where necessary. For example:

- **Working capital** – Are sufficient funds available to deliver the customer expectations, e.g. to fund initial changes to gain emotional acceptance during the deployment?
- **Contracts and licenses** – Have all necessary contract and licence transfers been arranged?
- **Funding** – Is funding available for the supporting systems to manage the service, e.g. CMS and related licences?
- **Intellectual property** – Has the full range of IP, its ongoing ownership and usage has been addressed, including:
  - Software developed by one of the parties
  - Documentation such as user manuals?

4.4.5.2 Preparation for build, test and deployment

Before authorizing the build and test stage, the Service Design and the release design must be validated against the requirement for the new or changed service offering. This should result in constructive feedback on the Service Design. Record, track and measure any risks and issues against the services, service assets and CIs within the service package, SLP, SDP or release package. Prioritize the issues and actions to ensure they can be resolved in a timely manner. Finally, produce a validation report and associated results ready for service evaluation.

An independent evaluation of the service and release design uses the validation report and results (see 4.6.5). This evaluation checks that the change to the services or service offering will deliver the predicted outcomes, i.e. the service expected by the user or customer. If there are issues, an interim evaluation report is prepared. This report lists the deviations from the SDP, a risk profile and recommendations for Change Management. If there are deviations in the service level requirements then the service package, SLP or SAC may be changed (via Change Management) and action should be taken to modify the proposed service release and related changes. Successful completion of the evaluation of the Service Design baseline ensures that service release build and test starts with a stable, baselined and approved design.

For some releases the Service Transition Manager may need to assign individuals or establish a team of competent people to execute the plans. If individuals are not dedicated there is risk that they may be diverted to work on other projects. Such risks need to be mitigated as they are often the cause of delays.

On most occasions, the introduction of a technology-enabled service requires training for the release, deployment, build and test teams. The training needs of these groups will be at different levels. Recognition of the different skill sets, capabilities and competencies within the various groups is a useful prerequisite in identifying the necessary training. In specifying the training programme, the number of people that require training needs to be determined, and the way the knowledge can be provided needs to be considered. While the need for training differs from release to release, the impact of training can be significant. For example if support staff are spread around many locations, specific training, automated mechanisms, such as e-learning or computer-based training (CBT) solutions over the internet or intranet, may become an attractive proposition.

Examples of training needs include:

- Interpreting the Service Design documentation and plans
- Use of support tools, e.g. for central release staff
- Changes in health and safety requirements
- Changes in security policies and procedures
- Technical training
- Service Management and process training, e.g. new build procedure for new configuration item type.

4.4.5.3 Build and test

During the build and test stages, the common services and infrastructure need to be managed carefully since they can significantly affect the build and test of a technology enabled service and its underlying technology infrastructure. Key aspects that need to be managed during the activities to build and test a service or service offering are:

- Usage of the build and test environments
- Standardization and integration aspects
- Management of the configurations:
  - During the build and test activities, e.g. version control, baseline management, control of inputs and outputs from a build or test stage
  - Recording the complete record of the build so that it can be rebuilt if required
  - Maintaining evidence of testing, e.g. test results and test report
  - Controlling access rights to physical and technology components, e.g. setting parameters
  - Checking that security requirements are met
  - Verification activities, e.g. prerequisites are met before a build or test begins
  - Managing environmental issues, e.g. space, cooling, power, fire precautions, accessibility and safety measures
  - Preparing and controlling the service release ready for promotion to the next environment
  - Promoting or handing over the service release to the next stage or team.

Configuration baselines of the controlled environments and the release package before and after an installation, build or deployment are recorded in the CMS to provide a restore point. The configuration information also needs to be updated to reflect the receipt and implementation of a release unit or the complete release package to a deployment group or target environment. The definitive version of the release package (approved in service release test) must be placed in the DML even where the release package consists only of documentation for a hardware upgrade. The release package must always be taken from the DML to deploy to the Service Operations readiness, service acceptance and live environments.

Release and build documentation

Procedures, templates and guidance should be used to enable the release team to take service assets and products from internal and external suppliers and build an integrated release package efficiently and effectively.

Procedures and documents for purchasing, distributing, installing, moving and controlling assets and components that are relevant to acquiring, building and testing a release include:

- Contract and agreements (e.g. for ordering new equipment or software)
- Purchase requests and ordering
- Request fulfilment
- Goods inwards and delivery
- Health and safety guidelines
- Security policies and procedures
- Leasing agreements
- Intellectual property rights/digital rights
- Support agreements
- Procedures for:
  - Managing service and infrastructure configurations
  - Distributing and installing software
  - Distributing, translating and converting data and information
  - Delivering, installing and moving equipment
  - Cleansing data and media
  - Disposing of documentation, media and equipment
  - Building, commissioning and decommissioning test environments, infrastructures and facilities
  - Publishing knowledge, information and data
  - Validation and testing
  - Change Management
- Service Asset and Configuration Management
- Acceptance and authorization
- Documenting licence agreements and licence headings together with 'proof of licence'.

'Proof of licence' is what a court will accept as proof of a legal entity having a licence. Each software manufacturer in general states the requirements for their proof of licence, so no hard and fast rules can be given here. As a general principle, proof of licence requires some form of evidence directly from the software manufacturer. There is a spectrum of types of evidence for having a proof of licence. Typical examples include:

- Printed licence confirmation documents from software manufacturers (with security features)
- Electronic licence confirmation documents from software manufacturers held on controlled-access websites

- Certificates of authenticity (COAs), which are typically engraved, or with other security features. These may be loose pieces of paper, pieces of paper pasted onto manual covers, labels glued onto equipment, labels printed or glued on retail boxes.

The proposed solution should be documented to enable knowledge gathered during the build and test stages to be handed over to the Service Operations and Continual Service Improvement to be retained for future releases. It is important that the information is ordered and maintained in a systematic manner as during the build and test activities updates to the documentation will be required. The documentation includes:

- Roles and responsibilities
- Process descriptions and procedures
- Support and operations manuals, service desk scripts etc.
- Communications, training and knowledge transfer deliverables
- User manuals with work instructions
- Service information
- Business context and marketing information
- Service catalogue, SLA and supporting documentation:
  - Hardware and software information
  - Logical and physical architectural overview
  - Detailed technical descriptions and references
- Technical information
- Service Management and operations plans
- Business continuity planning details
- Index of documentation for the service and release – baselined.

Acquire and test input configuration items and components

Configuration items and components (e.g. services, service assets) are acquired from projects, suppliers, partners and development groups. To prevent the acquisition of unknown and potentially risky components for a build it is essential to use CIs that have achieved a certain quality level or components from a catalogue of standard components that have been previously assessed, tested and authorized for use in specific conditions. Otherwise a change will need to be raised to assess the component and either incorporate it into the standards catalogue or accept it as a one-off exception for this release.

The acquisition activities include:

- Interfacing with procurement processes to acquire the components (or with internal production departments if supplied in-house)
- Capturing and recording:
  - New or updated service assets and CIs through SACM
  - Receipt of components
  - Delivery, change and release documentation from the supplier
- Checking, monitoring and reporting the quality of incoming CIs and service components
- Ensuring that proof of licence can be demonstrated where required
- Initiating action if quality is different from expectation, and assess the likely impact of this on the transition
- Updating status of configuration items in SACM, e.g. to indicate that they are ready to be released into the next stage or rejected.

Verification activities to check the components destined for a release package or build include:

- Establishing that all items are bona fide, and have genuinely been ordered or commissioned
- Standard labelling and naming conventions have been applied as specified in the design specifications for the CIs and service components
- Recording externally acquired items and checking these against their delivery and release documentation
- Checking that:
  - Developed products and service components have successfully passed appropriate documented quality reviews
  - All software is as expected and no malicious additions are included (e.g. software items that could contain viruses)
  - All amendments to previous versions or configuration baselines have been authorized by Change Management and no other amendments have been included – this may require a configuration audit and comparison facilities to check against the desired configuration
  - All definitive items have been added to the DML and correctly recorded in the CMS
  - Rejection/return of components is adequately controlled and documented.

Issues, non-conformance, known errors and deviations reports about the quality of service components and any risks should be passed to the relevant stakeholders, e.g. quality assurance, CSI, Service Design.

Release packaging

Build management procedures, methodologies, tools and checklists should be applied to ensure that the release package is built in a standard, controlled and reproducible way in line with the solution design defined in the Service Design Package. As a release package progresses towards production it may need to be rebuilt. For example: if a newer version of a CI or component needs to be incorporated quickly to fix errors; if the documentation needs to be updated.

The key activities to build a release package are:

- Assemble and integrate the release components in a controlled manner to ensure a reproducible process.
- Create the build and release documentation including:
  - Build, installation and test plans, procedures and scripts
  - Details of how to monitor and check the quality of the release and how to recognize and react to problems
  - The automated or manual processes and procedures required to distribute, deploy and install the release into the target environment (or remove it as necessary)
  - Procedures to back out release units or remediate a change should a release fail
  - Procedures for tracking and managing software licences and digital rights.
- Install and verify the release package.
- Baseline the contents of the release package.
- Send a service notification to inform relevant parties that the release package is available for installation and use.

If testing of a release package is successful, the release and the contents of the release package are placed under the control of Configuration Management, baselined and verified against the release design and release package definition. From this point all changes to the release package are managed through Change Management, e.g. to fix an error in testing. If at any step the testing of a release package does not complete successfully, reassessment and rescheduling of the release is managed through Change Management.

Build and manage the test environments

Effective build and test environment management is essential to ensure that the builds and tests are executed in a repeatable and manageable manner. Inadequate control of these environments means that unplanned changes can compromise the testing activities and/or cause significant re-work. Dedicated build environments should be established for assembling and building the components for controlled test and deployment environments.

Preparation of the test environments includes building, changing or enhancing the test environments ready to receive the release.

An IT service is, on most occasions, built from a number of technology resources or management assets. In the build phase, these different blocks, often from different suppliers, are installed and configured together to create the solution as designed. Standardization facilitates the integration of the different building blocks to provide a working solution and service.

Automating the installation of systems and application software onto servers and workstations reduces the dependencies on people and streamlines the procedures. Depending on the release and deployment plans, the installation may be performed in advance (for example, if equipment is being replaced) or it may have to occur in situ in the live environment.

The physical infrastructure elements, together with the environment in which they will operate, need to be tested appropriately. Part of the testing may be to test the replication of the infrastructure solution from one environment to another. This gives a better guarantee that the rollout to the production environment will be successful.

Test environments must be actively maintained and protected using Service Management best practices. For any significant change to a service, the question should be asked (as it is for the continued relevance of continuity and capacity plans): 'If this change goes ahead, will there need to be a consequential change to the test data?' During the build and test activities, operations and support teams need to be kept fully informed and involved as the solution is built to facilitate a structured transfer from the project to the operations team.

4.4.5.4 Service testing and pilots

The testing activities are coordinated through test management, which plans and controls the testing execution that is described in section 4.5. Testing aims to build confidence in the service capability prior to final acceptance during pilot or early life support. It will be based on the test strategy and model for the service being changed.

The test criteria reflect the anticipated conditions in which the service is expected to operate and deliver benefit. However, these surrounding circumstances may change, and in many modern situations such change is almost inevitable and often unpredictable. These changes and their impact on service testing and acceptance must be observed, understood and documented. Their consequences need to be expressed in terms of changed Acceptance Criteria and updates to the service package, including the SLP. This will need the collaboration and input of the business, customers and other affected stakeholders, which may well include suppliers and operations. The Service Designer will be involved in making any amendments since this knowledge may assist in building in additional and relevant flexibility to designs of future new or changed services.

An example of tests that can be executed during release and deployment is shown in Figure 4.22. Further details of these tests are described in section 4.5 on validation and testing. In practice, the test types overlap the different levels of testing to provide a full range of testing across the service life.

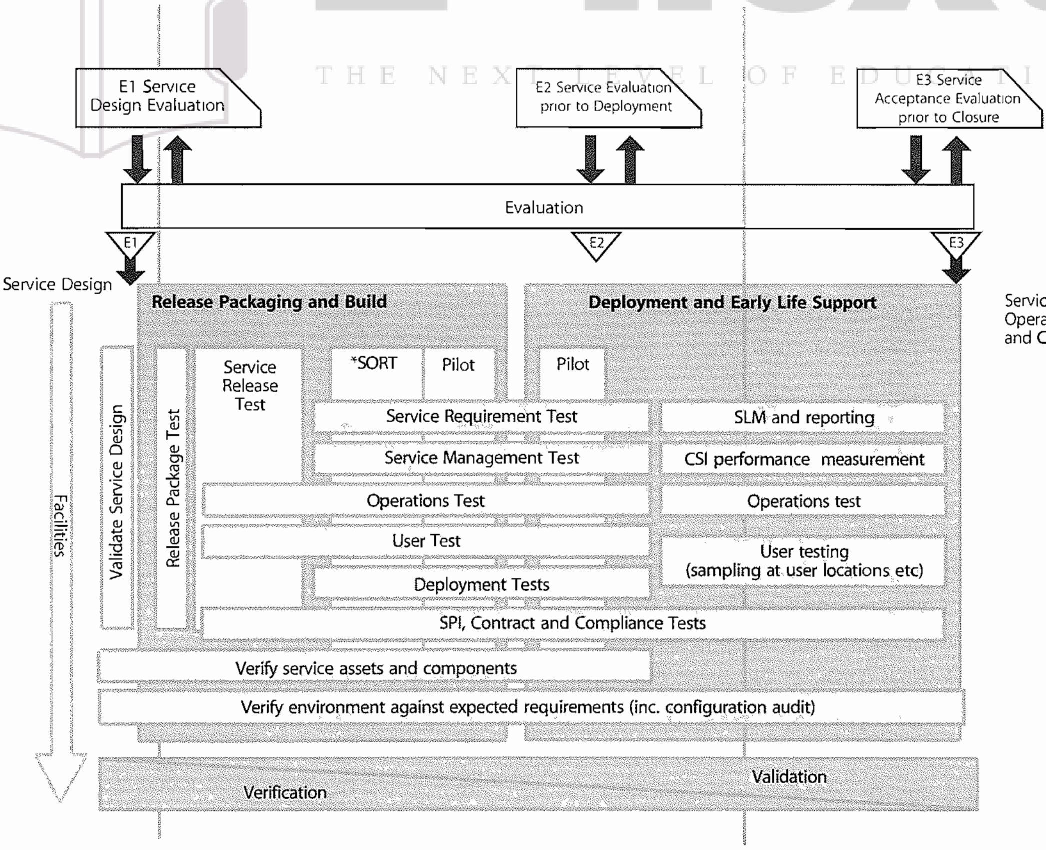
A service release test checks that the service components can be integrated correctly and that the release can be installed, built and tested in the target environment.

Service Operations readiness testing ensures that a service and its underlying application and technology infrastructure can be transferred into the production environment in a controlled manner. It provides a level of confidence that the new or changed service will provide the level of service specified in the service requirements and service level requirements. However, it is too early to finalize the SLA at this point. The SLA is finalized in the pilot or more usually in early life support before the Service Transition is closed. The service operational readiness test aims to:

- Determine whether a service and its underlying service assets can be released into the production environment, the first time and for subsequent deployments
- Ensure that the business processes, customers, users and service provider interfaces (SPIs) are capable of using the service properly
- Ensure that the service teams are capable of operating the service and using the Service Management systems properly.

Tests that are conducted as part of service operational readiness test include:

- **Deployment readiness test** – to ensure that the deployment processes, procedures and systems can


E1 Service Design Evaluation
E2 Service Evaluation prior to Deployment
E3 Service Acceptance Evaluation prior to Closure
Evaluation
E1
E2
E3
Service Design
Release Packaging and Build
Deployment and Early Life Support
Service Operation and CSI
Facilities
Validate Service Design
Release Package Test
Service Release Test
*SORT
Pilot
Pilot
Service Requirement Test
SLM and reporting
Service Management Test
CSI performance measurement
Operations Test
Operations test
User Test
User testing (sampling at user locations etc)
Deployment Tests
SPI, Contract and Compliance Tests
Verify service assets and components
Verify environment against expected requirements (inc. configuration audit)
Verification
Validation

*Service Operational Readiness Test

Figure 4.22 Example of service testing through Service Transition

deploy, install, commission and decommission the release package and resultant new or changed service in the production/deployment environment

- **Service Management test** – to ensure that the service performance can be measured, monitored and reported in production
- **Service Operations test** – to ensure that the service teams will be able to operate the service in production
- **Service level test** – to ensure that the new or changed service will deliver the service level requirements
- **User test** – to ensure that users can access and use the new or changed service, e.g. they have access to the updated service catalogue and contact details for the service desk
- **Service provider interface test** – to ensure that interfaces to the service are working
- **Deployment verification test** – to ensure that the service capability has been correctly deployed for each target deployment group or environment.

Service rehearsals

One testing method is to simulate as much of the service as possible in a service rehearsal (sometimes referred to as 'model office'). A service rehearsal is a simulation of as much of the service as possible in an extensive and widely participatory practice session. It is the ultimate stage of internal testing, the last stage before any public live running. This is like a 'dress rehearsal' of a play, setting out all the elements – costume, lighting etc. – in a last private run-through of the performance. It can deliver significant benefits by establishing errors and unworkable procedures before they impact the business in live operation. However, they are complex, time consuming and relatively expensive to prepare, deliver and document. A careful and deliberate balance is therefore required between the anticipated costs and the risk damage profile that they could prevent.

A service rehearsal takes place just before deployment of the service; if held too early there is a significant chance that the environment, technology, people and legislation into which the service is being released will change and invalidate the results. If too close to the declared release date any issues found will not be addressed before the service goes live.

The objectives of the service rehearsal include:

- Confirmation that all stakeholders have been identified and are committed to operating or using the service – if not this will be evidenced through lack of players for roles within the service rehearsal
- Ensure that all stakeholders have processes and procedures in place and are ready to receive process and resolve incidents, problems and changes relating to the new or changed service
- Testing the effectiveness of 'mistake-proofing' included within the service procedures. (Mistake proofing, often referred to by the Japanese term 'Poca Yoke', is about introducing advance warnings of user mistakes or bad practice and where possible introducing steps in the procedures to prevent these mistakes – such as electrical switch interlocks, and check-sum digits in data entry.) While testing can check how a service reacts for predicted user error, the service rehearsal will encourage unforeseen behaviour and establish how that behaviour affects the service's ability to deliver the required benefits.

The service rehearsal requires adequate representation from all stakeholders, with commitment to providing staff for – typically – a full day rehearsal for a new or significantly changed service. It is often beneficial to involve 'ordinary' representatives of the stakeholder community, not those with previous experience or knowledge of the service. Typical mistakes will be more likely to come from typical users – those who have been involved in design and development will find it impossible to 'unlearn' and will be coloured by their expectations of service behaviour.

The focus of a service rehearsal is typically on one day of actual rehearsal, but successful delivery of a service rehearsal involves more stages, including preparation and analysis, mirroring the Plan–Do–Check–Act cycle. Typical stages for a service rehearsal would include the following activities.

Plan – prepare for the day

Request for a service rehearsal – the project or service implementation teams consider that a service rehearsal would be appropriate and trigger the process with a request.

Tasks include the following:

- Appoint a rehearsal manager who gathers all relevant information.
- Identify key and secondary processes.
- Identify all stakeholders and their contact information.
- Produce initial rehearsal guide – the script to be followed.
- Establish and document typical examples of incidents, service requests, capacity and availability issues and

other events that will need to be handled when the service is live.

- Produce documentation to allow the simulation, processing, tracking and analysis of the expected scenarios.
- Identify all stakeholders, supplier and service provider personnel who need to be involved and ensure their commitment, through direct funding, internal commitment etc.
- Create detailed scripts – in collaboration with customer or account manager.
- Invite all stakeholders to planning and preparation meetings and briefings (could be by documentation, e-mail, Webinars etc. if physical briefings are not practicable.)

Do – deliver the rehearsal

Hold meetings to:

- Introduce the objectives, documents, involvement, recording etc.
- Walkthrough the scenarios and scripts to establish authenticity of the approach at a detailed level
- Carry out the rehearsal, i.e. let the players deliver the script and observe the processing of key events and elements, e.g. follow an incident through from occurrence to loggings, diagnosis, resolution, recovery and closure.

Check – document the day

Tasks include:

- Analysing and evaluating the results of the rehearsal and determining the implications
- Producing a written test report on the rehearsal, with recommendations, e.g. re-work the service before deployment
- Recording identified errors, issues and risks.

Act – take action following the rehearsal

Considering the results from the rehearsal, the options will be:

- Declare service to have passed without serious concern.
- OR consider that the service is not suitable for progressing at this stage and refer back to Service Design and/or Service Transition for re-work and rescheduling. (It may occasionally be that service rehearsal shows that the actual environment within which the service is expected to function is different enough from expectation to prevent acceptable behaviour from the service in reality – this might require rethink and revision at the Service Strategy and/or business process level.)
- Review and close the service rehearsal, providing improvement ideas to CSI, SD and ST management as appropriate.

Pilots

Pursuing the theatrical analogy seen in service rehearsal, if the service rehearsal is the 'dress rehearsal' – the last practice before being seen by the public – then the pilot is the 'off Broadway' run of a play. It is done for real and in public, but for a small audience only and with the expectation of further (hopefully minor) polishing of the performance, script, scenery and effects. Conducting a pilot is easier to control as it is deployed to a smaller environment/user base.

A pilot sets out to detect if any elements of the service do not deliver as required and to identify gaps/issues in Service Management that put the service and/or the customer's business and assets at risk. It does not need to cover all service and system functionality, but will focus on the areas of risk and perform enough of the service to determine if it will work sufficiently well in deployment. It aims to ensure that the service capability supports delivery of the service requirements and service level requirements. As far as possible it should check that the service utilities are fit for purpose and the warranties are fit for use.

Establish clear objectives for the pilot implementation such as:

- To establish metrics and provide confidence that the predicted performance and service levels will be met
- To evaluate the actual benefits and costs achieved during the pilot against the Business Case
- To create acceptance of new processes and ways of working within the user base, service provider and suppliers
- To identify, assess and mitigate some of the risks associated with a full deployment.

As there are likely to be design changes and improvements that need to be built into the release before full deployment, it is important to agree how these will be funded up front. It is also important to ensure that there is a common understanding about how the pilot implementation will be signed off.

During the pilot the release and deployment team should:

- Be ready to invoke contingency/recovery procedures
- Involve key people that will be involved in the full deployment

- Ensure that people involved in the pilot are trained and that they understand their new/changed role and responsibilities
- Document necessary operational and support procedures, information and training material that can not be adequately simulated in a test environment
- Establish the viability of training and support documentation and modify where necessary
- Establish customer, user and stakeholder interaction with the service in real-time situations, e.g. with real business decisions being made
- Capture appropriate metrics in order to compare to the service performance model
- Establish additional criteria that may need to be met before full deployment starts
- Determine the likely level of service support and Service Management resources that will be required and resolve any issues
- Discover and fix issues and errors early and fix many of them before final deployment. This includes the less critical minor irritations and eccentricities of a service that would not necessarily cause non-acceptance but do significantly reduce the emotional acceptance of the service among the user community
- Document improvements and where appropriate incorporate them into plans for full deployment.

When the release has been in use for a sufficient period during a pilot it is important to check that the service is capable of delivering the requirements of the customer, user and the Service Design as well as the predicted outcomes (although not all these will be realized at this point).

If the pilot is of sufficient length, it may be appropriate to conduct an independent evaluation to compare the actual vs predicted service capability and performance (specified in the Service Design) on behalf of the stakeholders, users and customers. This evaluation includes a risk assessment on whether the service will continue to deliver the service requirements, e.g. service levels and warranties.

The outputs from a successfully delivered service pilot will include:

- New or changed service and capability that have been tested and evaluated
- Pilot test report and results
- A report generated by the evaluation function, which is passed to Change Management and which comprises: an updated risk profile, deviations report, recommendation
- Key stakeholder agreement that the release is ready for a full deployment
- Demonstrated benefits of the service (within agreed tolerance levels)
- Confirmation that the deployment team has tested the deployment process and accepts the cost model, deployment model and metrics to be used for monitoring during deployment and early life support
- Target deployment groups in different geographical locations accepting the service release and committing to the deployment plans, particularly groups with different cultures and languages.

4.4.5.5 Plan and prepare for deployment

The planning and preparation activities prepare the deployment group for deployment. This is an opportunity to prepare the organization and people for organizational change; see section 5.2. The overall approach to planning the deployment is described in release and deployment planning (see paragraph 4.4.5.1). During the actual deployment stage the detailed implementation plan is developed. This includes assigning individuals to specific activities. For example a specific individual may be assigned to deliver training for a training activity on the deployment plan.

The entry criteria for planning and preparing a target deployment group or environment include:

- Deployment stakeholders are sufficiently confident in the service release to deploy the release, own their aspects of deployment and they are committed to the deployment (see section 5.2).
- Senior management, customers, the business and service provider teams accept the deployment costs, management, organization and people implications of the release as well as any organization, function and process changes.

An example of the deployment activities that apply to the deployment for a target group is shown in Figure 4.23.

Preparing for deployment includes assessing each deployment group's readiness to receive and implement a release package, identifying gaps that need to be filled and planning the activities required to deploy, transfer or decommission/retire services or service assets. It will also include transferring a service or a service unit as well as move and disposal activities.

Assessment

Although the deployment assessment should be conducted early, it should be revisited periodically. The results of this assessment are fed into detailed implementation planning for the target deployment group.

The readiness assessment for a deployment group identifies:

- Issues and risks in delivering the current services that may affect the deployment. The kinds of risk include:
  - Lack of dedicated internal resources and external supplier resources
  - Lack of training, skills and awareness
  - Unplanned or late change in requirements
- Anticipated impacts, e.g. on the organizational structure, environment for the new or changed services, direct customers and users, partners, suppliers
- Gaps that need to be filled.

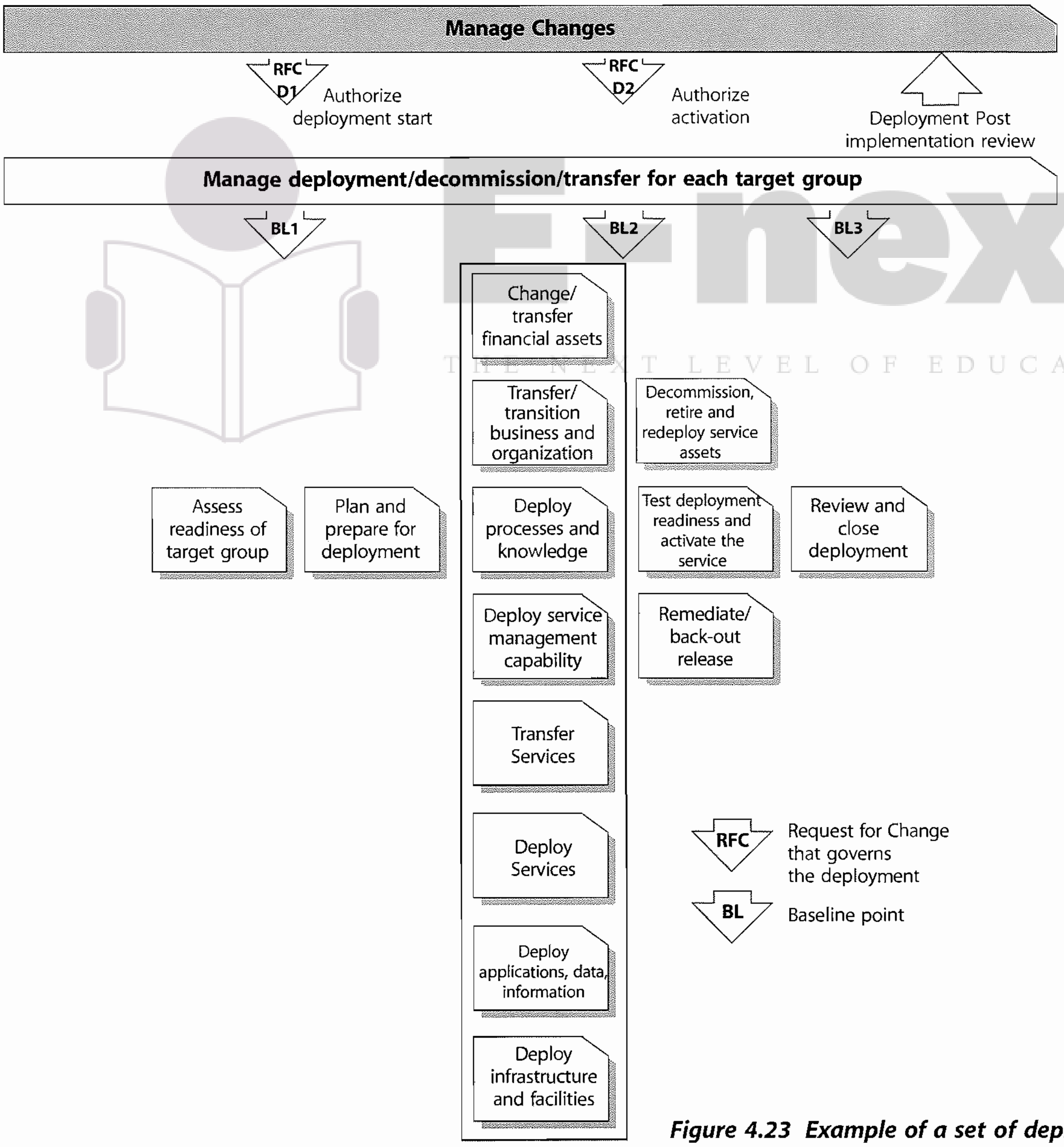

Manage Changes
RFC D1
Authorize deployment start
RFC D2
Authorize activation
Deployment Post implementation review
Manage deployment/decommission/transfer for each target group
BL1
BL2
BL3
Change/ transfer financial assets
Transfer/ transition business and organization
Decommission, retire and redeploy service assets
Assess readiness of target group
Plan and prepare for deployment
Deploy processes and knowledge
Test deployment readiness and activate the service
Review and close deployment
Deploy service management capability
Remediate/ back-out release
Transfer Services
Deploy Services
Deploy applications, data, information
Deploy infrastructure and facilities
RFC
Request for Change that governs the deployment
BL
Baseline point

Figure 4.23 Example of a set of deployment activities

The aspects to assess include:

- Financial aspects and assets:
  - Current and required working capital
  - Establishing new or changed contracts, licences, IPR and digital rights
- Issues and risks in delivering the current services that may affect the deployment
- Applicable health, safety, security and environmental regulations, supplier and procurement aspects
- Current capability of the business customers and users to use and gain value from the new or changed service
- Current service, service capability and resources used including:
  - Service structure
  - Service dynamics
  - Service metrics and reports, including warranties and service levels achieved
- Current Service Management capability and resources:
  - Differences from the prerequisites for deployment, e.g. inadequate licensing arrangements, network bandwidth
  - Current operations and support resources, e.g. tools, people
  - Support resources and workloads as there may be a significant increase in the number of incidents per user that can stretch the resources for managing incidents, problems and fixes
  - Performance reports and improvement plans
  - Ability to predict and track the actual incident and problem volumes during deployment; this may require updating asset or user records with the date and time of installation or deployment to enable trend analysis
- Identifying requirements to tailor the new or changed service or underlying solution, e.g. processes, procedures, work instructions
- Organizational readiness:
  - Role, resource and skills gap analysis
  - Training needs analysis
  - Ability to assign competent individuals to the required roles
  - Motivation and empowerment – does the current organization and culture encourage the application of the required skills? Is there the right leadership and commitment?
  - Assess the readiness of customers, users, service provider staff and other stakeholders such as suppliers, partners
- Aspects relating to applications, information and data:
  - Access to application, information and data
  - Accessing secret, restricted or confidential documents and data
  - Knowledge and experience in using the application – users and support staff
- Infrastructure and facilities:
  - Difficult access, e.g. located high up in a building without appropriate lifting equipment (elevator or crane, etc.); city centre with restricted parking; remote locations
  - Intermediate and final storage and stores for definitive hardware and media
  - IT equipment space and capacity requirements such as:
    - size and equipment footprints
    - power requirements and circuit-breaker ratings
    - uninterruptible power supply (UPS) and generator loadings
    - temperature and humidity requirements
    - heat outputs and air-conditioning requirements
    - door clearance and engineering access requirements
    - cabling requirements
  - Electromagnetic interference (EMI) and radio frequency interference (RFI) requirements
  - Air quality requirements
  - Weight and false floor loadings
  - Network considerations
  - Equipment health, safety, security and environmental requirements.

Develop plans and prepare for deployment

Planning for a specific deployment includes assigning specific resources to perform deployment and early life support activities. While developing these plans, identify and assess risks specific to this deployment group by using the service model to identify business and service critical assets that have the highest risk of causing disruption. The activities include:

- Risk mitigation plans
- Developing transfer/transition, upgrade, conversion, disposal, retirement plans
- Logistics and delivery planning:
  - The service assets and components for deployment, establishing how and when they will be delivered, and confirmation that delivery has been successfully achieved and recorded

   - Site preparation in accordance with applicable health, safety, security and environmental regulations and requirements
- Tailoring processes, procedures and knowledge, e.g. language translation, time frame adjustments
- Knowledge transfer and training stakeholders in how to use, benefit, manage, support and operate the new or changed service:
   - Identify essential and potential recipients of training (such as customer, users, ITSM, service desk, support, operations, deployment teams, projects)
   - Update of service desk with knowledge of the target deployment group and their environment
- Communicating to the people involved:
   - About the changes and the expected benefits
   - How the change affects the organization and staff
- Making any changes in emergency of continuity plans and procedures
- Mobilizing the Service Operations and support organization
- Mobilizing users to be ready to use the service
- Additional activities identified from the assessment.

The next step is to verify the detailed deployment plans, perform any deployment readiness tests and raise an RFC to be authorized through the Change Management process. The service is then ready for deployment.

4.4.5.6 Perform transfer, deployment and retirement

The following activities provide an example of the different aspects that will be performed in the order specified on the deployment plan.

Transfer financial assets

Changes and transfers of financial assets need to be completed as part of deployment. This will include but is not constrained by the following:

- Any changes in supplier financial agreements and charges
- Purchase or transfer of annual support and maintenance costs including systems to manage the service, e.g. CMS
- New licence costs and renewals
- Annual disaster recovery contracts with third parties
- Provision or transfer of working capital
- Transfer of intellectual property.

Transfer/transition business and organization

Transfer of a business unit, service or service unit will involve change to the organization itself. The subject of organizational change is addressed in Chapter 5. Activities that need to be performed include:

- Finalize organization structure, roles and responsibilities.
- Communicate change in organization, roles and responsibilities.
- Ensure that people adapt to and adopt new practices. This requires good communication of the consequences and requirements of the deployed service, e.g. best use of resources to deliver the message; understanding personal and group concerns; and ensuring messages to diverse and related groups are consistent and appropriate.
- Engender, at the very least, acceptance and preferably active support of the changes imposed on people.
- Ensure that people understand the continuity plans and procedures.

When the change includes a transfer of service provider, e.g. new outsourcing, insourcing, change of outsourced provider, then some specific elements need to be considered, e.g. organizational change, quick wins to avoid confusion and higher staff turnaround.

Competent people with the right skills are required to perform the deployment, operate and manage the new or changed service in the business, customer and service provider organization. The related activities include:

- Recruit staff with appropriate skills. Rather than developing new skills for existing staff, it may be more efficient to recruit new staff who already have the required skills. This may be in addition to existing staff, or may require the replacement of some staff with inappropriate skills, with more relevant staff for the revised circumstances of the new service.
- Identify existing people (e.g. staff, suppliers, users) with appropriate skills, moving or re-allocating people as necessary. For the skills required to actually deploy the new or changed service, temporary secondment, or even overtime, may be the most efficient approach.
- Consider outsource/contract resources to provide the required skills. This is similar to seconding internal staff, but in this case buying the temporarily required skills from external providers where they already exist. If skills are needed longer term, a requirement to pass those skills on to permanent (or longer term) staff can be useful.

- Provide training. Manage the training logistics, coordination, setup, communications, registration, delivery and evaluation activities including users and Service Operations teams.
- Execute the knowledge transfer plan and track progress to completion.
- Evaluate competence of new and changed staff and other people.

Deploy processes and materials

Deploy or publish the processes and materials ready for people involved in the business and service organization change, e.g. users and Service Operations teams that need to execute the new or changed processes. The materials may include policies, processes, procedures, manuals, overviews, training products, organizational change products etc.

Training people to use new processes and procedures can take time, particularly for a global deployment to thousands of people.

Deploy Service Management capability

Deploy new or changed processes, systems and tools to the service provider teams responsible for Service Management activities. Check that everyone is competent and confident to operate, maintain and manage the service in accordance with the service model and processes. Remove or archive redundant services and assets, e.g. processes, procedures and tools.

During deployment monitor the service against the service model and performance standards as far as possible.

Transfer service

Transferring a service will also involve organizational change described earlier in this section. The issues around transferring a service and the activities that need to be performed include:

- Reviewing the service performance, issues and risks, by performing some service tests and a service evaluation prior to the transfer
- Configuration auditing of service assets and configurations
- Finalizing service catalogue (add or remove the service) and related information
- Sending a service notification to communicate the change to relevant stakeholders.

When the change includes a transfer of service provider, e.g. new outsourcing, insourcing, change of outsourced provider, then some specific elements need to be considered that include:

- Managing contract changes
- Managing changes to existing agreements
- Updating contract details and information in the SKMS
- Transferring ownership of service assets and configuration items, remembering to update the CMS.

Deploy service

Deploy the service release and carry out the activities to distribute and install the service, supporting services, applications, data, information, infrastructure and facilities. These will include:

- Distributing and delivering the service and service components at the correct location and time
- Building, installing and configuring the services and service components with any converted or new data and information
- Testing the system and services according to the installation and acceptance tests and producing the installation and test reports
- Recording any incidents, unexpected events, issues or deviations from the plans
- Correcting any deviations that are outside the design limitations and constraints.

Decommissioning and service retirement

Some specific aspects need to be considered for decommissioning and retiring services and service assets. For example the procedures for retiring, transferring (e.g. to another budget holder) or redeploying service assets need to take into account any security, confidentiality, licensing, environmental or other contractual requirements. This includes:

- Removing deployed copies of software and data from retired hardware; failure to do this may result in licence contravention or in staff using unsupported software
- Identifying licences and other assets which can be redeployed; software being retired from use in one area may well remain in active use elsewhere
- Disposing of equipment according to environmental policies and procedures

- Moving assets that can be redeployed to secure storage areas if required. If the assets being retired are remaining in use elsewhere, especially for hardware, the released assets may serve a useful role as spare equipment to be retained in asset stores for speedy redeployment in the event of failures.

Records of retirement, transfer and disposal should be maintained and used to update other information such as licence information.

Remove redundant assets

A comprehensive understanding of the assets used by a retired service needs to be gained and managed. With a full understanding any redundant assets can be identified and removed, therefore potentially saving licence fees, liberating capacity and preventing accidental use. Failure to develop and properly perform these activities can result in:

- Wasted disk space and licences
- Overpayment of licence and maintenance fees
- Removal of assets associated with the redundant service but also used by other services, therefore causing incidents within those services, e.g. common software components and network elements.

As part of the clean-up activities it is important to delete or archive redundant data, information and records related to the previous service or products. The full scope and scale of a service or service asset needs to be considered, and this should extend to the following areas:

- Support contracts with third party suppliers, as changes in likely usage may require renegotiation of contracts.
- In-house second/third level support staff with specialist knowledge may no longer require that knowledge. This may require re-assessment of their role, level of payment, retention etc. and opportunities for redeployment may be identified.
- Service desk workload may be affected.
- Records within the knowledge base relating to the decommissioned components may need to be archived and deleted.

4.4.5.7 Verify deployment

When the deployment activities are complete, it is important to verify that users, Service Operations, other staff and stakeholders are capable of using or operating the service. The tests should specifically verify that:

- The service, service assets and service capability/resources are in place, e.g. by performing an audit such as a configuration audit of the deployed baseline against the as-planned baseline
- Updates to documentation and information are completed, e.g. service catalogue, contracts, agreements, contact details
- Communications, orientation and learning materials are ready to distribute to stakeholders, Service Operations and users
- All roles are assigned to individuals/organizations
- People and other resources are prepared to operate and use the new or changed service or service capability in normal, emergency and disaster situations
- People have access to the information necessary to use, operate or support the service
- The measurement and reporting systems are established to measure performance of the service and underlying resources.

This is a good point to gather feedback on the deployment process to feed into future improvements, e.g. using satisfaction surveys.

Report any issues and incidents and take corrective actions as necessary.

Successful confirmation of the deployment verification triggers the initiation and launch of early life support for the deployment group.

4.4.5.8 Early life support

Early life support (ELS) provides the opportunity to transition the new or changed service to Service Operations in a controlled manner and establish the new service capability and resources. An example of the ELS activities is shown in Figure 4.24.

In Service Design, the stakeholders will have agreed the entry and exit criteria from early life support but it may be necessary to finalize the performance targets and exit criteria early in this stage. This can help to understand the deployment verification process and set customer and stakeholder expectations about the handover of the service to Service Operations.

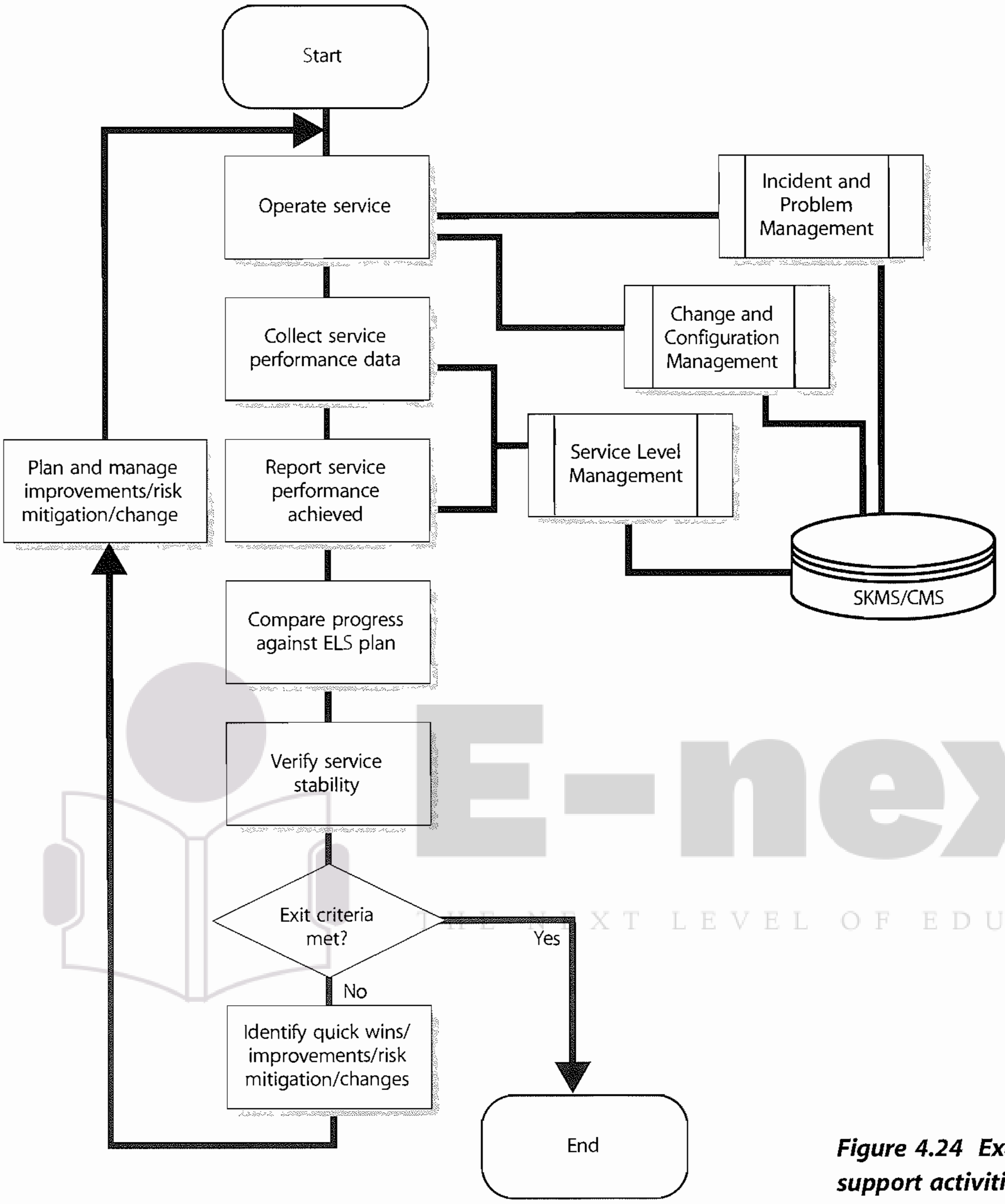

Start
Operate service
Collect service performance data
Report service performance achieved
Compare progress against ELS plan
Verify service stability
Exit criteria met?
Yes
No
Identify quick wins/ improvements/risk mitigation/changes
End
Plan and manage improvements/risk mitigation/change
Incident and Problem Management
Change and Configuration Management
Service Level Management
SKMS/CMS

Figure 4.24 Example of early life support activities

ELS provides appropriate resources to resolve operational and support issues quickly, centrally and locally, to ensure that the users can use the service to support their business activities without unwarranted disruption. The deployment teams should analyse where users and support resources will experience issues and problems, perhaps based on previous experience; for example, clarification about:

- Role assignments, roles and responsibilities
- Financial and funding arrangements
- Procurement and request fulfilment
- Security policies and procedures
- Raising incidents and change requests
- Escalation procedures
- Complaints procedure
- Using diagnostics tools and aids
- Software licensing rules.

During ELS, the deployment team implements improvements and resolves problems that help to stabilize the service. The Continual Service Improvement publication provides relevant information on measurement and service improvements. The deployment resources will gradually back out from providing the additional support as the users and service teams become familiar with the changes and the incidents and risks reduce.

Metrics for the target deployment group or environment measure service performance, performance of the Service Management and operations processes and teams and the number of incidents and problems by type. The deployment team's aim is to stabilize the service for the

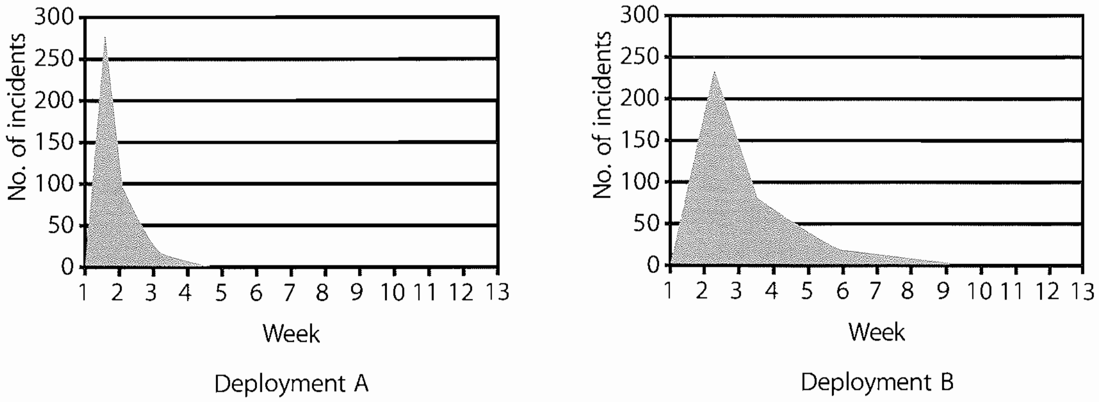


Figure 4.25 Illustration of the benefits of targeted early life support

target deployment group or environment as quickly and effectively as possible. An example of a deployment performance graph is shown in Figure 4.25.

Variation in performance between different deployment groups and service units should be analysed and lessons learned from one deployment used to improve subsequent deployments.

The example shown in Figure 4.25 shows the number of incidents for two branches of a retail organization that have the same number of users and the same deployment schedule. In deployment A the incident levels have reduced faster. On further investigation the Service Transition manager discovered that the team responsible for Deployment A was more competent at training users and transferring knowledge to the service desk so that they could help users to be more effective more quickly.

During ELS, the deployment team should ensure that the documentation and knowledge base are updated with additional diagnostics, known errors, workarounds and frequently asked questions. The team should also resolve any knowledge transfer or training gaps.

At agreed milestones in early life support, it is important to assess the issues and risks, particularly those that impact the handover schedule and costs. Service Transition monitors the performance of the new or changed service in early life support until the exit criteria are achieved. These include when:

- Users can use the service effectively and efficiently for their business activities
- Service owners and process owners are committed to manage and operate the service in accordance with the service model, performance standards and processes
- Service delivery is managed and controlled across any service provider interfaces
- Consistent progress is being made towards delivering the expected benefits and value at each milestone in early life support
- Service levels and service performance standards are being consistently achieved without unexpected variation before formal handover to Service Operations
- SLAs are finalized and signed off by senior management and customers
- Unexpected variations in the performance of the service and customer assets such as changes in residual risks are monitored, reported and managed appropriately
- Checking that training and knowledge transfer activities are completed by obtaining positive confirmation from the target audience. This may be in the form of competency tests
- The service release, the SLA, other agreements and any contractual deliverables are signed off.

4.4.5.9 Review and close a deployment

When reviewing a deployment the following activities should be included:

- Capture experiences and feedback on customer, user and service provider satisfaction with the deployment, e.g. through feedback surveys.
- Highlight quality criteria that were not met.
- Check that any actions, necessary fixes and changes are complete.
- Review open changes and ensure that funding and responsibility for open changes are agreed before handover.
- Review performance targets and achievements, including resource use and capacity such as user accesses, transactions and data volumes.

- Make sure there are no capability, resource, capacity or performance issues at the end of the deployment.
- Check that any problems, known errors and workarounds are documented and accepted by the customers/business and/or suppliers.
- Review the Risk Log and identify those that impact Service Operations and support. Address risks or agree action such as moving the risks to the Service Transition Risk Log.
- Check that redundant assets have been removed.
- Check that the service is ready for transition from early life support into Service Operations.

Each deployment should consider whether any relevant issues have been detected that should be passed through to CSI, such as:

- Feedback on the deployment model and plan
- Errors in procedures detected
- 'Near misses' where things could have gone wrong in foreseeable circumstances or where intervention was required
- Incorrect data or information in relevant records
- Incident and problems caused by deployment
- Problems with updating records.

Deployment is completed with a handover of the support for the deployment group or target environment to Service Operations.

A post implementation review of a deployment is conducted through Change Management.

4.4.5.10 Review and close Service Transition

In order to finalize that a Service Transition is completed, there should be a formal review carried out that is appropriate to the scale and magnitude of the change. A review of the Service Transition should include:

- Checking that all transition activities completed, e.g. documentation and information is captured, updated, secured, archived
- Checking that accurate metrics were captured.

Independent evaluation of the service release uses the outputs from deployment. This evaluation checks the actual performance and outcomes of the new or changed service against the predicted performance and outcomes, i.e. the service expected by the user or customer. An evaluation report (see 4.6.6) is prepared that lists the deviations from the SP/SLP/SDP, a risk profile and recommendations for Change Management. If there are deviations in the service level requirements then the service package, SLP or SAC may need to change (via Change Management, in agreement with the customer representative and other stakeholders). Successful completion of the evaluation ensures that the service can be formally closed and handed over to Service Operations and CSI.

A transition report should be produced that summarizes the outcomes. As part of producing such a report a post transition workshop could be held involving all parties as a 'lessons learned' exercise. Lessons learned and improvements are fed into Change Management for a post implementation review and into Continual Service Improvement for future transitions.

4.4.6 Triggers, input and output, and inter-process interfaces

The release process commences with receipt of an approved RFC to deploy a production-ready release package. Deployment commences with receipt of an approved RFC to deploy a release package to a target deployment group or environment, e.g. business unit, customer group and/or service unit.

The inputs are:

- Authorized RFC
- Service package, SLP
- SDP, including service model and SAC
- IT service continuity plan and related business continuity plan
- Service Management and operations plans and standards
- Technology and procurement standards and catalogues
- Acquired service assets and components and their documentation
- Build models and plans
- Environment requirements and specifications for build, test, release, training, disaster recovery, pilot and deployment
- Release policy and release design from Service Design
- Release and deployment models including template plans
- Exit and entry criteria for each stage of release and deployment.

The outputs are:

- Release and deployment plan
- Completed RFCs for the release and deployment activities
- Service notification
- Updated service catalogue with the relevant information about the new or changed service

- New tested service capability and environment including SLA, other agreements and contracts, changed organization, competent and motivated people, established business and Service Management processes, installed applications, converted databases, technology infrastructure, products and facilities
- New or changed Service Management documentation
- Service package that defines the requirements from the business/customer for the service
- SLP that defines the service level requirements, e.g. hours of service, business critical services, data and periods, service level targets
- SLA, underpinning OLAs and contracts
- Service model that describes the structure and dynamics of how the service is operated and managed
- New or changed service reports
- Tested continuity plans
- Complete and accurate configuration item list with an audit trail for the CIs in the release package and also the new or changed service and infrastructure configurations
- Service capacity plan that is aligned to the relevant business plans
- Deployment ready release package (baselined) – for future deployments
- Service Transition Report.

Deployment is completed with a handover of the new or changed service to operations on successful completion of the post implementation review of the deployment conducted within Change Management.

4.4.7 Information management

Throughout the deployment process, appropriate records will be created and maintained. As configuration items are successfully deployed, the CMS will be updated with information such as:

- New or changed configuration items
- Relationships between requirements and test cases
- Installation/build plans
- Logistics and delivery plans
- Validation and test plans, evidence and reports
- New or changed locations and users
- Status updates (e.g. from allocated to live)
- Change in ownership of assets
- Licence holding.

Other data and information will also be captured and recorded within the broader service knowledge management system. This could include:

- Deployment information, history of the deployment itself, who was involved, timings etc.
- Training records, typically held by HR in many organizations, but relating to ITSM staff the responsibility for their update will logically rest with ITSM also.
- Access rules and levels
- Known errors. Typically a new or changed service will be introduced with identified errors, which while not according to the original Service Design specification are nonetheless minor enough in nature to be acceptable in live operation. These may well be under active investigation and resolution by the service builders, or may be considered acceptable. In either case the errors will be deployed into the live error database as an element of the deployment of the live service. This information will be available through the SKMS to the service desk who will then be able to link incidents reported against these known errors.

As part of the clean-up activities it is important to delete or archive redundant records related to the previous service or products.

4.4.8 Key performance indicators and metrics

4.4.8.1 Customers or business

Indicators include:

- Variance from service performance required by customers (minimal and reducing)
- Number of incidents against the service (low and reducing)
- Increased customer and user satisfaction with the services delivered
- Decreased customer dissatisfaction – service issues resulting from poorly tested or untested services increases the negative perception on the service provider organization as a whole.

4.4.8.2 Service providers

Indicators include:

- Reduced resources and costs to diagnose and fix incidents and problems in deployment and production
- Increased adoption of the Service Transition common framework of standards, re-usable processes and supporting documentation
- Reduced discrepancies in configuration audits compared with the real world.

4.4.9 Challenges, critical success factors and risks

Challenges for release and deployment include:

- Developing standard performance measures and measurement methods across projects and suppliers
- Dealing with projects and suppliers where estimated delivery dates are inaccurate and there are delays in scheduling Service Transition activities
- Understanding the different stakeholder perspectives that underpin effective risk management for the change impact assessment and test activities
- Building a thorough understanding of risks that have impacted or may impact successful Service Transition of services and releases
- Encouraging a risk management culture where people share information and take a pragmatic and measured approach to risk.

Critical success factors include:

- The new or changed service capability and resources are built in the target environment or deployment group.
- The new or changed service has been tested against the Service Design.
- The service capability has been proved in a pilot deployment.
- Re-usable test models are developed that can be used for regression testing in future releases.

Risks to successful release and deployment include:

- Poorly defined scope and understanding of dependencies in earlier lifecycle stages leading to scope creep during release and deployment
- Using staff that are not dedicated to release and deployment activities, especially if the effort is a significant amount of their time
- Management:
    - Management incompetence
    - Inadequate corporate policies, e.g. security, software licensing
    - Inadequate adoption of management practices
    - Poor leadership
- Finances:
    - Shortage of finances
    - Delays move deployment into different financial year
    - Lack of clarity on funding for changes/fixes during transition
- Controls:
    - Lack of definition of the required controls leads to poorly evaluated and unauthorized changes, adversely affecting release and deployment plans
    - Difficulty tracking and managing software licences, e.g. due to complexity
    - Unexpected or changes in regulatory controls or licensing requirements
- Management of organizational change
    - Unclear expectations/objectives from customers, users, suppliers and other stakeholders
    - Cultural differences/misunderstandings
    - Human factors
    - With suppliers/partners
    - Poor communication
    - Organizational change impacts employee morale
    - People issues with infringement of personal data protection criteria
    - Personality clashes
    - Key personnel who have inadequate authority to fulfil their roles
    - Poor staff recruitment and selection procedures
    - Lack of clarity over roles and responsibilities
    - Vested interests creating conflict and compromising quality
    - Individual or group interests are given unwarranted priority
- Poor commitment and decision making
- Failure to obtain appropriate approval at the right time
- Indecision or late decision making
- Lack of operational support
- Inadequate or inaccurate information
- Health and safety compromised
- The time allowed for release and deployment – will it make or break the project?
- Suppliers/sourcing/partnering relationships during transition:
    - Failure of suppliers to meet contractual requirements; this could be in terms of quality, quantity, timescales or their own exposure to risk
    - Delays in contract negotiation
    - Organizational change impacts employee morale, employee and supplier performance
    - Data protection impacts data sharing
    - Shrinking resource pool from disaffected employees
- Governance issues:
    - Senior management commitment is missing in one or other of the organizations

  - The supplier management function is not mature or is non-existent
  - Changes in work practices and procedures adversely affect one or other of the organizations
- Inadequate 'back-out' or 'contingency' plan if sourcing/partnering fails
- Application/technical infrastructure risks:
  - Inadequate design
  - Professional negligence
  - Human error/incompetence
  - Infrastructure failure
  - Differences/dependencies in infrastructure/ applications
  - Increased dismantling/decommissioning costs
  - Safety being compromised
  - Performance failure (people or equipment)
  - Breaches in physical security/information security
  - Unforeseen barriers or constraints due to infrastructure.

4.5 SERVICE VALIDATION AND TESTING

The underlying concept to which Service Testing and Validation contributes is quality assurance – establishing that the Service Design and release will deliver a new or changed service or service offering that is fit for purpose and fit for use. Testing is a vital area within Service Management and has often been the unseen underlying cause of what was taken to be inefficient Service Management processes. If services are not tested sufficiently then their introduction into the operational environment will bring a rise in:

- Incidents, since failures in service elements and mismatches between what was wanted and what was delivered impact on business support
- Service desk calls for clarification, since services that are not functioning as intended are inherently less intuitive causing a higher support requirement
- Problems and errors that are harder to diagnose in the live environment
- Costs, since errors are more expensive to fix in production than if found in testing
- Services that are not used effectively by the users to deliver the desired value.

4.5.1 Purpose, goal and objectives

The purpose of the Service Validation and Testing process is to:

- Plan and implement a structured validation and test process that provides objective evidence that the new or changed service will support the customer's business and stakeholder requirements, including the agreed service levels
- Quality assure a release, its constituent service components, the resultant service and service capability delivered by a release
- Identify, assess and address issues, errors and risks throughout Service Transition.

The goal of Service Validation and Testing is to assure that a service will provide value to customers and their business.

The objectives of Service Validation and Testing are to:

- Provide confidence that a release will create a new or changed service or service offerings that deliver the expected outcomes and value for the customers within the projected costs, capacity and constraints
- Validate that a service is 'fit for purpose' – it will deliver the required performance with desired constraints removed
- Assure a service is 'fit for use' – it meets certain specifications under the specified terms and conditions of use
- Confirm that the customer and stakeholder requirements for the new or changed service are correctly defined and remedy any errors or variances early in the service lifecycle as this is considerably cheaper than fixing errors in production.

4.5.2 Scope

The service provider takes responsibility for delivering, operating and/or maintaining customer or service assets at specified levels of warranty, under a service agreement. Service Validation and Testing can be applied throughout the service lifecycle to quality assure any aspect of a service and the service providers' capability, resources and capacity to deliver a service and/or service release successfully. In order to validate and test an end-to-end service the interfaces to suppliers, customers and partners are important. Service provider interface definitions define the boundaries of the service to be tested, e.g. process interfaces and organizational interfaces.

Testing is equally applicable to in-house or developed services, hardware, software or knowledge-based services. It includes the testing of new or changed services or service components and examines the behaviour of these in the target business unit, service unit, deployment group or environment. This environment could have aspects outside the control of the service provider, e.g. public networks, user skill levels or customer assets.

Testing directly supports the release and deployment process by ensuring that appropriate levels of testing are performed during the release, build and deployment activities. It evaluates the detailed service models to ensure that they are fit for purpose and fit for use before being authorized to enter Service Operations, through the service catalogue. The output from testing is used by the evaluation process to provide the information on whether the service is independently judged to be delivering the service performance with an acceptable risk profile.

4.5.3 Value to business

Service failures can harm the service provider's business and the customer's assets and result in outcomes such as loss of reputation, loss of money, loss of time, injury and death. The key value to the business and customers from Service Testing and Validation is in terms of the established degree of confidence that a new or changed service will deliver the value and outcomes required of it and understanding the risks.

Successful testing depends on all parties understanding that it cannot give, indeed should not give, any guarantees but provides a measured degree of confidence. The required degree of confidence varies depending on the customer's business requirements and pressures of an organization.

4.5.4 Policies, principles and basic concepts

4.5.4.1 Inputs from Service Design

A service is defined by a service package that comprises one or more service level packages (SLPs) and re-usable components, many of which themselves are services, e.g. supporting services. The service package defines the service utilities and warranties that are delivered through the correct functioning of the particular set of identified service assets. An SLP provides a definitive level of utility or warranty from the perspective of outcomes, assets and patterns of business activity (PBA) of customers. It is therefore a key input to test planning and design.

The design of a service is related to the context in which a service will be used (the categories of customer asset). The attributes of a service characterize the form and function of the service from a utilization perspective. These attributes should be traceable to the predicted business outcomes that provide the utility from the service. Some attributes are more important than others for different sets of users and customers, e.g. basic, performance and excitement attributes. A well-designed service provides a combination of these to deliver an appropriate level of utility for the customer.

The Service Design Package defines the agreed requirements of the service, expressed in terms of the service model and Service Operations plan that provide key input to test planning and design. Service models are described further in the Service Strategy publication.

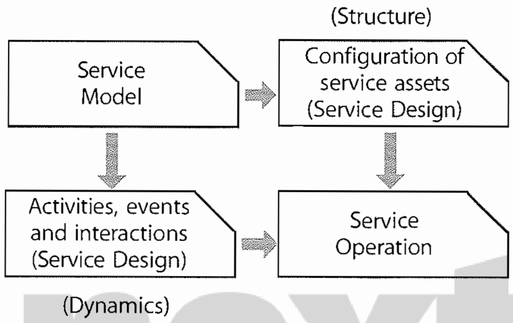

(Structure)
Service Model
Configuration of service assets (Service Design)
Activities, events and interactions (Service Design)
Service Operation
(Dynamics)

Figure 4.26 Service models describe the structure and dynamics of a service

The service model (Figure 4.26) describes the structure and dynamics of a service that will be delivered by Service Operations, through the Service Operations plan. Service Transition evaluates these during the validation and test stages.

Structure is defined in terms of particular core and supporting services and the service assets needed and the patterns in which they are configured. As the new or changed service is designed, developed and built, the service assets are tested and verified against the requirements and design specifications: is the service asset built correctly?

For example, the design for managed storage services must have input on how customer assets such as business applications utilize the storage, the way in which storage adds value to the applications, and what costs and risks the customer would like to avoid. The information on risks is of particular importance to service testing as this will influence the test coverage and prioritization.

Service models also describe the dynamics of creating value. Activities, flow of resources, coordination, and interactions describe the dynamics (see Figure 4.27). This includes the cooperation and communication between

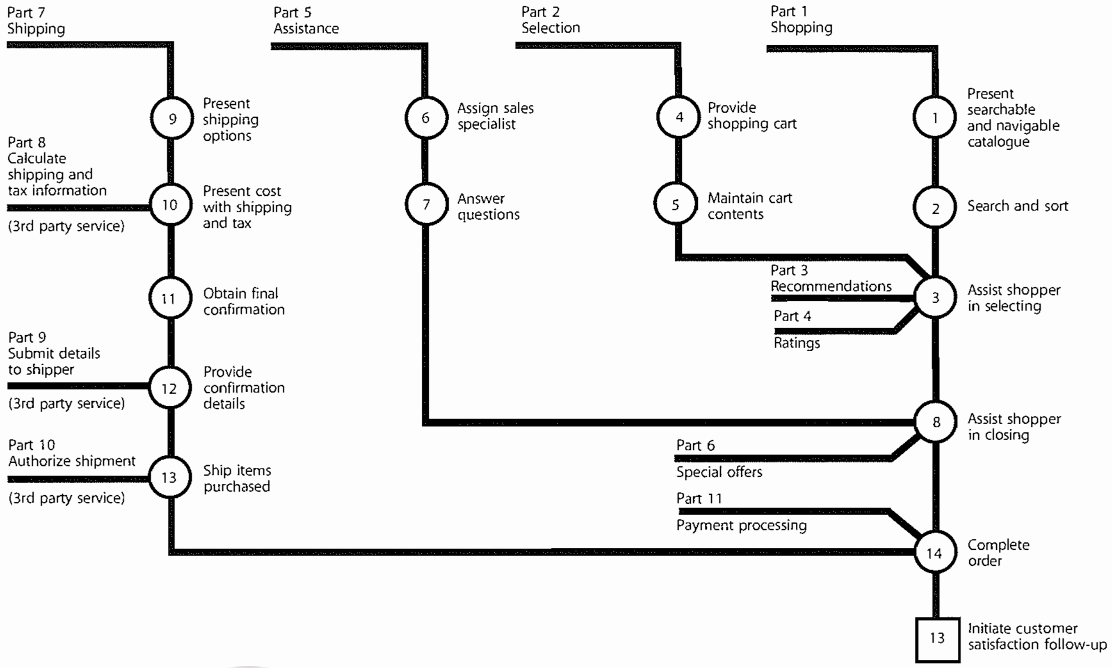


Figure 4.27 Dynamics of a service model

service users and service agents such as service provider staff, processes or systems that the user interacts with, e.g. a self-service menu. The dynamics of a service include patterns of business activity, demand patterns, exceptions and variations.

Service Design uses process maps, workflow diagrams, queuing models, and activity patterns to define the service models. As Service Transition evaluates the detailed service models to ensure they are fit for purpose and fit for use it is important to have access to these models to develop the test models and plans.

The Service Design package defines a set of design constraints (Figure 4.28) against which the service release and new or changed service will be developed and built.

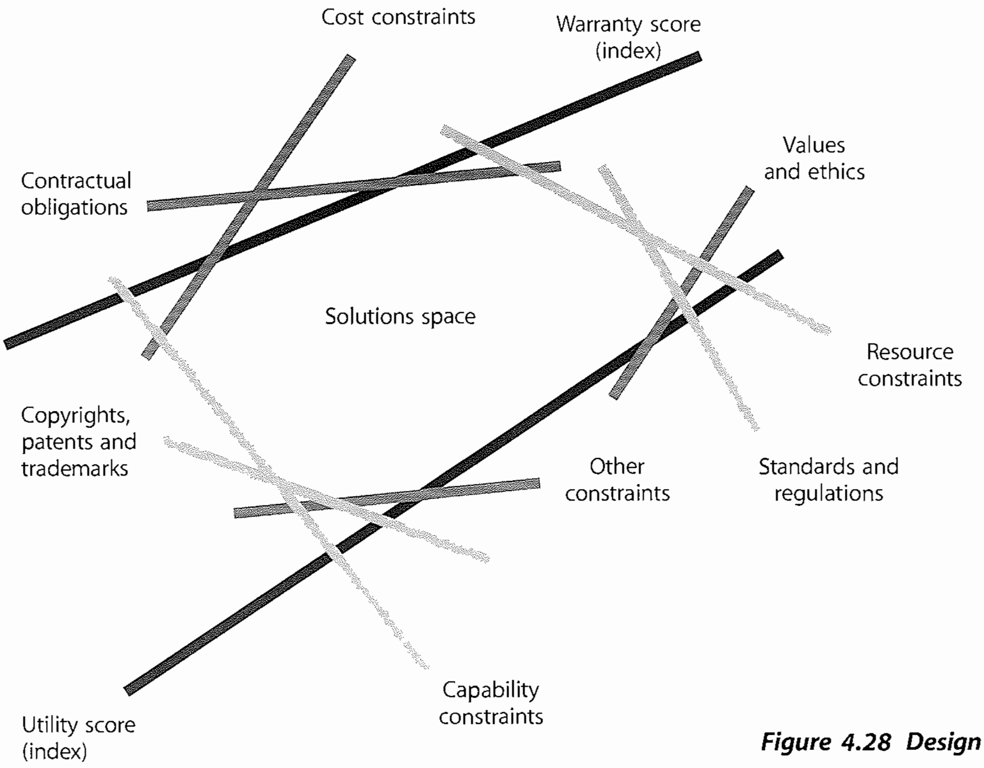


Figure 4.28 Design constraints of a service

Validation and testing should test the service at the boundaries to check that the design constraints are correctly defined and particularly if there is a design improvement to add or remove a constraint.

4.5.4.2 Service quality and assurance

Service assurance is delivered though verification and validation, which in turn are delivered through testing (trying something out in conditions that represent the final live situation – a test environment) and by observation or review against a standard or specification.

Validation confirms, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled. Validation in a lifecycle context is the set of activities ensuring and gaining confidence that a system or service is able to accomplish its intended use, goals and objectives.

The validation of the service requirements and the related service Acceptance Criteria begins from the time that the service requirements are defined. There will be increasing levels of service validation testing performed as a service release progresses through the service lifecycle.

Verification is confirmation, through the provision of objective evidence, that specified requirements have been fulfilled, e.g. a service asset meets its specification.

Early in the service lifecycle, validation confirms that the customer needs, contracts and service attributes, specified in the service package, are translated correctly into the Service Design as service level requirements and constraints, e.g. capacity and demand limitations. Later in the service lifecycle tests are performed to assess whether the actual service delivers the required levels of service, utilities and warranties. The warranty is an assurance that a product or service will be provided or will meet certain specifications. Value is created for customers if the utilities are fit for purpose and the warranties are fit for use (Figure 4.29). This is the focus of service validation.

4.5.4.3 Policies

Policies that drive and support Service Validation and Testing include service quality policy, risk policy, Service Transition policy, release policy and Change Management policy.

Service quality policy

Senior leadership will define the meaning of service quality. Service Strategy discusses the quality perspectives that a service provider needs to consider. In addition to service level metrics, service quality takes into account the positive impact of the service (utility) and the certainty of impact warranty. The Service Strategy publication outlines four quality perspectives:

- Level of excellence
- Value for money
- Conformance to specifications
- Meeting or exceeding expectations.

One or more, if not all four, perspectives are usually required to guide the measurement and control of Service Management processes. The dominant perspective will influence how services are measured and controlled, which in turn will influence how services are designed and operated. Understanding the quality perspective will influence the Service Design and the approach to validation and testing.

Risk policy

Different customer segments, organizations, business units and service units have different attitudes to risk. Where an organization is an enthusiastic taker of business risk, testing will be looking to establish a lower degree of confidence than a safety critical or regulated organization might seek. The risk policy will influence control required

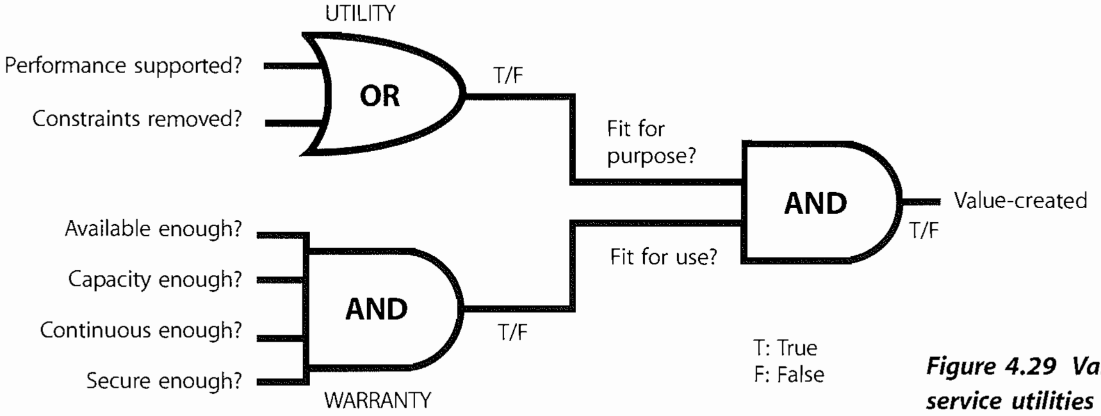


Figure 4.29 Value creation from service utilities and warranties

through Service Transition including the degree and level of validation and testing of service level requirements, utility and warranty, i.e. availability risks, security risks, continuity risks and capacity risks.

Service Transition policy

See Chapter 3.

Release policy

The type and frequency of releases will influence the testing approach. Frequent releases such as once-a-day drive requirements for re-usable test models and automated testing.

Change Management policy

The use of change windows can influence the testing that needs to be considered. For example if there is a policy of 'substituting' a release package late in the change schedule or if the scheduled release package is delayed then additional testing may be required to test this combination if there are dependencies.

The testing policy will reflect the requirements from Service Strategy. Examples of policy statements include:

- Test library and re-use policy. The nature of IT Service Management is repetitive, and benefits greatly from re-use. The service test management role within an organization should take responsibility for creating, cataloguing and maintaining a library of test models, test cases, test scripts and test data that can be re-used. Projects and service teams need to be motivated and incentivized to create re-usable test assets and re-use test assets.
- Integrate testing into the project and service lifecycle. This helps to detect and remove functional and non-functional defects as soon as possible and reduces the incidents in production.
- Adopt a risk-based testing approach aimed at reducing risk to the service and the customer's business.
- Engage with customers, stakeholders, users and service teams throughout the project and service lifecycle to enhance their testing skills and capture feedback on the quality of services and service assets.
- Establish test measurements and monitoring systems to improve the efficiency and effectiveness of Service Validation and Testing Continual Service Improvement.
- Automate using automated testing tools and systems, particularly for:
  - Complex systems and services, such as geographically distributed services, large-scale infrastructures and business critical applications
  - Where time to change is critical, e.g. if there are tight deadlines and a tendency to squeeze testing windows.

4.5.4.4 Test strategy

A test strategy defines the overall approach to organizing testing and allocating testing resources. It can apply to the whole organization, a set of services or an individual service. Any test strategy needs to be developed with appropriate stakeholders to ensure there is sufficient buy-in to the approach.

Early in the lifecycle the service validation and test role needs to work with Service Design and service evaluation to plan and design the test approach using information from the service package, SLPs, SDP and the interim evaluation report. The activities will include:

- Translating the Service Design into test requirements and test models, e.g. understanding combinations of service assets required to deliver a service as well as the constraints that define the context, approach and boundaries to be tested
- Establishing the best approach to optimize the test coverage given the risk profile and change impact and resource assessment
- Translating the service Acceptance Criteria into entry and exit criteria at each level of testing to define the acceptable level of margin for errors at each level
- Translating risks and issues from the impact, resource and risk assessment on the related RFC for the SDP/service release into test requirements.

It is also vital to work with Project Managers to ensure that:

- Appropriate test activities and resources are included in Project Plans
- Specialist testing resources (people, tools, licences) are allocated if required
- The project understands the mandatory and optional testing deliverables
- The testing activities are managed, monitored and controlled.

The aspects to consider and document in developing the test strategy and related plans are shown below. Some of the information may also be specified in the Service Transition plan or other test plans and it is important to structure the plans so that there is minimal duplication.

Test strategy contents

- Purpose, goals and objectives of service testing
- Context

- Applicable standards, legal and regulatory requirements
- Applicable contracts and agreements:
  - Service Management policies, processes and standards
  - Policies, processes and practices applicable to testing
- Scope and organizations:
  - Service provider teams
  - Test organization
  - Third parties, strategic partners, suppliers
  - Business units/locations
  - Customers and users
- Test process:
  - Test management and control – recording, progress monitoring and reporting
  - Test planning and estimation, including cost estimates for service planning, resources, scheduling
  - Test preparation, e.g. site/environment preparation, installation prerequisites
  - Test activities – planning, performing and documenting test cases and results
- Test metrics and improvement
- Identification of items to be tested:
  - Service package
  - Service level package
  - SDP – service model (structure and dynamics), solution architecture design
- Service Operation plan
- Service Management Plans:
  - Critical elements where business priorities and risk assessment suggest testing should concentrate
  - Business units, service units, locations where the tests will be performed
- Service provider interfaces
- Approach:
  - Selecting the test model
  - Test levels
  - Test approaches, e.g. regression testing, modelling, simulation
  - Degree of independence for performing, analysing and evaluating tests
  - Re-use – experience, expertise, knowledge and historical data
  - Timing, e.g. focus on testing individual service assets early vs testing later when the whole service is built
  - Developing and re-using test designs, tools, scripts and data
  - Error and change handling and control
  - Measurement system
- Criteria:
  - Pass/fail criteria
  - Entry and exit criteria for each test stage
  - For stopping or re-starting testing activities
- People requirements:
  - Roles and responsibilities including approval/rejection (these may be at different levels, e.g. rejecting an expensive and long running project typically requires higher authority than accepting it as planned)
  - Assigning and scheduling training and knowledge transfer
  - Stakeholders – service provider, suppliers, customer, user involvement
- Environment requirements:
  - Test environments to be used, locations, organizational, technical
  - Requirements for each test environment
  - Planning and commissioning of test environment
- Deliverables:
  - Mandatory and optional documentation
  - Test plans
  - Test specifications – test design, test case, test procedure
  - Test results and reports
  - Validation and qualification report
  - Test summary reports.

4.5.4.5 *Test models*

A test model includes a test plan, what is to be tested and the test scripts that define how each element will be tested. A test model ensures that testing is executed consistently in a repeatable way that is effective and efficient. The test scripts define the release test conditions, associated expected results and test cycles.

To ensure that the process is repeatable, test models need to be well structured in a way that:

- Provides traceability back to the requirement or design criteria
- Enables auditability through test execution, evaluation and reporting
- Ensures the test elements can be maintained and changed.

Examples of test models are illustrated in Table 4.10.

Table 4.10 Examples of service test models

Test model	Objective/target deliverable	Test conditions based on
Service contract test model	To validate that the customer can use the service to deliver a value proposition.	Contract requirements. Fit for purpose, fit for User criteria.
Service requirements test model	To validate that the service provider can/has delivered the service required and expected by the customer	Service requirements and Service Acceptance Criteria.
Service level test model	To ensure that the service provider can deliver the service level requirements, and service level requirements can be met in the production environment, e.g. testing the response and fix time, availability, product delivery times, support services.	Service level requirements, SLA, OLA.
Service test model	To ensure that the service provider is capable of delivering, operating and managing the new or changed service using the 'as-designed' service model that includes the resource model, cost model, integrated process model, capacity and performance model etc.	Service model.
Operations test model	To ensure that the Service Operations teams can operate and support the new or changed service/service component including the service desk, IT operations, application management, technical management. It includes local IT support staff and business representatives responsible for IT service support and operations. There may be different models at different release/test levels, e.g. technology infrastructure, applications.	Service model, Service Operations standards, processes and plans.
Deployment release test model	To verify that the deployment team, tools and procedures can deploy the release package into a target deployment group or environment within the estimated timeframe. To ensure that the release package contains all the service components required for deployment, e.g. by performing a configuration audit.	Release and deployment design and plan.
Deployment installation test model	To test that the deployment team, tools and procedures can install the release package into a target environment within the estimated timeframe.	Release and deployment design and plan.
Deployment verification test model	To test that a deployment has completed successfully and that all service assets and configurations are in place as planned and meet their quality criteria.	Tests and audits of 'actual' service assets and configurations.

As the Service Design phase progresses, the tester can use the emerging Service Design and release plan to determine the specific requirements, validation and test conditions, cases and mechanisms to be tested. An example is shown in Table 4.11.

Table 4.11 Service requirements, 1: improve user accessibility and usability

Validation reference	Validation condition	Test levels	Test case	Mechanism
1.1	20% improvement in user survey rating	1	M020	Survey
1.2	20% reduction in user complaints	1	M023	Process metrics
1.3	20% increase in use of self service channel	2	M123	Usage statistics
1.4	Help function available on front page of self service point application	3	T235	Functional test
1.5	Web pages comply with web accessibility standards	4 (Application)	T201	Usability test
1.6	10% increase in public self service points	4/5 Technical infrastructure	T234	Installation statistics
1.7	Public self-service points comply with standard IS1223	4/5 Technical infrastructure	T234	Compliance test

4.5.4.6 *Validation and testing perspectives*

Effective validation and testing focuses on whether the service will deliver as required. This is based on the perspective of those who will use, deliver, deploy, manage and operate the service. The test entry and exit criteria are developed as the Service Design Package is developed. These will cover all aspects of the service provision from different perspectives including:

- Service Design – functional, management and operational
- Technology design
- Process design
- Measurement design
- Documentation
- Skills and knowledge.

Service acceptance testing starts with the verification of the service requirements. For example, customers, customer representatives and other stakeholders who sign off the agreed service requirements will also sign off the service Acceptance Criteria and service acceptance test plan. The stakeholders include:

- Business customers/customer representatives
- Users of the service within the customer's business who will use the new or changed service to assist them in delivering their work objectives and deliver service and/or product to their customers
- Suppliers
- Service provider/service unit.

Business users and customer perspective

The business involvement in acceptance testing is central to its success, and is included in the Service Design package, enabling adequate resource planning.

From the business's perspective this is important in order to:

- Have a defined and agreed means for measuring the acceptability of the service including interfaces with the service provider, e.g. how errors or queries are communicated via a single point of contact, monitoring progress and closure of change requests and incidents
- Understand and make available the appropriate level and capability of resource to undertake service acceptance.

From the service provider's perspective the business involvement is important to:

- Keep the business involved during build and testing of the service to avoid any surprises when service acceptance takes place
- Ensure the overall quality of the service delivered into acceptance is robust, since this starts to set business perceptions about the quality, reliability and usability of the system, even before it goes live
- Deliver and maintain solid and robust acceptance test facilities in line with business requirements

- Understand where the acceptance test fits into any overall business service or product development testing activity.

Even when in live operation, a service is not 'emotionally' accepted by customer and user until they become familiar and content with it. The full benefit of a service will not be realized until that emotional acceptance has been achieved.

Emotional (non) acceptance

Southern US Steel Mill implemented a new order manufacturing service. It was commissioned, designed and delivered by an outside vendor. The service delivered was innovative and fully met the agreed criteria. The end result was that the company sued the vendor citing that the service was not usable because factory personnel (due to lack of training) did not know how to use the system and therefore emotionally did not accept it.

Testing is a situation where 'use cases', focusing on the usable results from a service can be a valuable aid to effective assessment of a service's usefulness to the business.

User testing – application, system, service

Testing is comprised of tests to determine whether the service meets the functional and quality requirements of the end users (customers) by executing defined business processes in an environment that, as closely as possible, simulates the live operational environment. This will include changes to the system or business process. Full details of the scope and coverage will be defined in the user test and user acceptance test (UAT) plans. The end users will test the functional requirements, establishing to the customer's agreed degree of confidence that the service will deliver as they require. They will also perform tests of the Service Management activities that they are involved with, e.g. ability to contact and use the service desk, response to diagnostics scripts, incident management, request fulfilment, change request management.

A key practice is to make sure that business users participating in testing have their expectations clearly set and realize that this is a test and to expect that some things may not go well. There is a risk that they may form an opinion too early about the quality of the service being tested and word may spread that the quality of the service is poor and should not be used.

Operations and service improvement perspective

Steps must be taken to ensure that IT staff requirements have been delivered before deployment of the service.

Operations staff will use the service acceptance step to ensure that appropriate:

- Technological facilities are in place to deliver the new or changed service
- Staff skills, knowledge and resource are available to support the service after go-live
- Supporting processes and resources are in place, e.g. service desk, second/third line support, including third party contracts, capacity and availability monitoring and alerting
- Business and IT continuity has been considered
- Access is available to documentation and SKMS.

Continual Service Improvement will also inherit the new or changed service into the scope of their improvement programme, and should satisfy themselves that they have sufficient understanding of its objectives and characteristics.

4.5.4.7 Levels of testing and test models

Testing is related directly to the building of service assets and products so that each one has an associated acceptance test and activity to ensure it meets requirements. This involves testing individual service assets and components before they are used in the new or changed service.

Each service model and associated service deliverable is supported by its own re-usable test model that can be used for regression testing during the deployment of a specific release as well as for regression testing in future releases. Test models help with building quality early into the service lifecycle rather than waiting for results from tests on a release at the end.

Levels of build and testing are described in the release and deployment section (paragraph 4.4.5.3). The levels of testing that are to be performed are defined by the selected test model.

Using a model such as the V-model (Figure 4.30) builds in Service Validation and Testing early in the service lifecycle. It provides a framework to organize the levels of configuration items to be managed through the lifecycle and the associated validation and testing activities both within and across stages.

The level of test is derived from the way a system is designed and built up. This is known as a V-model, which maps the types of test to each stage of development. The V-model provides one example of how the Service Transition levels of testing can be matched to corresponding stages of service requirements and design.

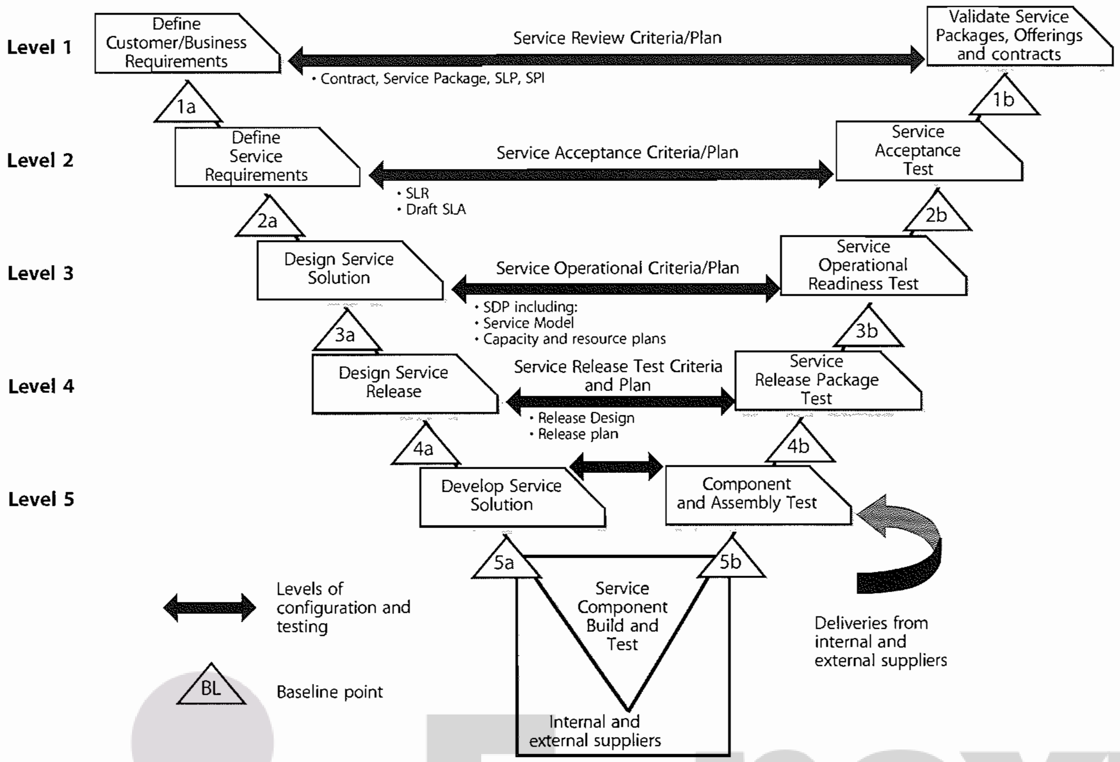

Level 1
Level 2
Level 3
Level 4
Level 5
Define Customer/Business Requirements
Service Review Criteria/Plan
• Contract, Service Package, SLP, SPI
Validate Service Packages, Offerings and contracts
1a
1b
Define Service Requirements
Service Acceptance Criteria/Plan
• SLR
• Draft SLA
Service Acceptance Test
2a
2b
Design Service Solution
Service Operational Criteria/Plan
• SDP including:
• Service Model
• Capacity and resource plans
Service Operational Readiness Test
3a
3b
Design Service Release
Service Release Test Criteria and Plan
• Release Design
• Release plan
Service Release Package Test
4a
4b
Develop Service Solution
Component and Assembly Test
5a
5b
Service Component Build and Test
Internal and external suppliers
Deliveries from internal and external suppliers
Levels of configuration and testing
BL
Baseline point

Figure 4.30 Example of service V-model

The left-hand side represents the specification of the service requirements down to the detailed Service Design. The right-hand side focuses on the validation activities that are performed against the specifications defined on the left-hand side. At each stage on the left-hand side, there is direct involvement by the equivalent party on the right-hand side. It shows that service validation and acceptance test planning should start with the definition of the service requirements. For example, customers who sign off the agreed service requirements will also sign off the service Acceptance Criteria and test plan.

4.5.4.8 Testing approaches and techniques

There are many approaches that can be combined to conduct validation activities and tests, depending on the constraints. Different approaches can be combined to the requirements for different types of service, service model, risk profile, skill levels, test objectives and levels of testing. Examples include:

- Document review
- Modelling and measuring – suitable for testing the service model and Service Operations plan
- Risk-based approach that focuses on areas of greatest risk, e.g. business critical services, risks identified in change impact analysis and/or service evaluation
- Standards compliance approach, e.g. international or national standards or industry specific standards
- Experience-based approach, e.g. using subject matter experts in the business, service or technical arenas to provide guidance on test coverage
- Approach based on an organization's lifecycle methods, e.g. waterfall, agile
- Simulation
- Scenario testing
- Role playing
- Prototyping
- Laboratory testing
- Regression testing
- Joint walkthrough/workshops
- Dress/service rehearsal
- Conference room pilot
- Live pilot.

In order to optimize the testing resources, test activities must be allocated against service importance, anticipated business impact and risk. Business impact analyses carried out during design for business and service continuity management and availability purposes are often very relevant to establishing testing priorities and schedules and should be available, subject to confidentiality and security concerns.

4.5.4.9 *Design considerations*

Service test design aims to develop test models and test cases that measure the correct things in order to establish whether the service will meet its intended use within the specified constraints. It is important to avoid focusing too much on the lower level components that are often easier to test and measure. Adopting a structured approach to scoping and designing the tests helps to ensure that priority is given to testing the right things. Test models must be well structured and repeatable to facilitate auditability and maintainability.

The service is designed in response to the agreed business and service requirements and testing aims to identify if these have been achieved. Service validation and test designs consider potential changes in circumstances and are flexible enough to be changed. They may need to be changed after failures in early service tests identify a change in the environment or circumstances and therefore a change on the testing approach.

Design considerations are applicable for service test models, test cases and test scripts and include:

- Business/Organization:
  - Alignment with business services, processes and procedures
  - Business dependencies, priorities, criticality and impact
  - Business cycles and seasonal variations
  - Business transaction levels
  - The numbers and types of users and anticipated future growth
  - Possible requirements due to new facilities and functionality
  - Business scenarios to test the end to end service
- Service architecture and performance:
  - Service Portfolio/structure of the services, e.g. core service, supporting and underpinning supplier services
  - Options for testing different type of service assets, utilities and warranty, e.g. availability, security, continuity
  - Service level requirements and service level targets
  - Service transaction levels
  - Constraints
  - Performance and volume predictions
  - Monitoring, modelling and measurement system, e.g. is there a need for significant simulation to recreate peak business periods? Will the new or changed service interface with existing monitoring and management tools?
- Service release test environment requirements
- Service Management:
  - Service Management models, e.g. capacity, cost, performance models
  - Service Operations model
  - Service support model
  - Changes in requirements for Service Management information
  - Changes in volumes of service users and transactions
- Application information and data:
  - Validating that the application works with the information/databases and technical infrastructure
  - Functionality testing to test the behaviour of the infrastructure solution and verify: i) no conflicts in versions of software, hardware or network components; and ii) common infrastructure services used according to the design
  - Access rights set correctly
- Technical infrastructure:
  - Physical assets – do they meet their specifications?
  - Technical resource capacity, e.g. storage, processing power, power, network bandwidth
  - Spares – are sufficient spares available or ordered and scheduled for delivery? Are hardware/software settings recorded and correct?

Aspects that generally need to be considered in designing service tests include:

- **Finance** – Is the agreed budget adequate, has spending exceeded budget, have costs altered (e.g. software licence and maintenance charge increases)?
- **Documentation** – Is all necessary documentation available or scheduled for production, is it practicable (sufficiently intuitive for the intended audience, available in all required languages), in correct formats such as checklists, service desk scripts?
- **Supplier** of the service, service asset, component – What are the internal or external interfaces?
- **Build** – Can the service, service asset or component be built into a release package and test environments?
- **Testable** – Is it testable with the resources, time and facilities available or obtainable?
- **Traceability** – What traceability is there back to the requirements?
- **Where and when** could testing take place? Are there unusual conditions under which a service might need to run that should be tested?
- **Remediation** – What plans are there to remediate or back out a release through the environments?

Awareness of current technological environments for different types of business, customer, staff and user is essential to maintaining a valid test environment. The design of the test environments must consider the current and anticipated live environment when the service is due for operational handover and for the period of its expected operation. In practice, for most organizations, looking more than six to nine months into the business or technological future is about the practical limit. In some sectors, however, much longer lead times require the need to predict further into the future, even to the extent of restricting technological innovation in the interests of thorough and expansive testing – examples are military systems, NASA and other safety critical environments.

Designing the management and maintenance of test data needs to address relevant issues such as:

- Separation of test data from any live data, including steps to ensure that test data cannot be mistaken for live data when being used, and vice versa (there are many real-life examples of live data being copied and used as test data and being the basis for business decisions e.g. desktop icons pointing at the wrong database)
- Data protection regulations – when live data is used to generate a test database; if information can be traced to individuals it may well be covered by data protection legislation, which for example may forbid its transportation between countries
- Backup of test data, and restoration to a known baseline to enable repeatable testing; this also applies to initiation conditions for hardware tests that should be baselined
- Volatility of test data and test environments, processes and procedures, which should be in place to quickly build and tear down the test environment for a variety of testing needs and so care must be taken to ensure that testing activities for one group do not compromise testing activities for another group
- Balancing cost and benefit – as test environments populated with relevant data are expensive to build and to maintain, so the benefits in terms of risk reduction to the business services must be balanced against the cost of provision. Also, how closely the test environment matches live production is a key consideration that needs to be weighed balancing cost with risk.

4.5.4.10 Types of testing

The following types of test are used to verify that the service meets the user and customer requirements as well as the service provider's requirements for managing, operating and supporting the service. Care must be taken to establish the full range of likely users, and then to test all the aspects of the service, including support and reporting.

Functional testing will depend on the type of service and channel of delivery. Functional testing is covered in many testing standards and best practices (see Further information).

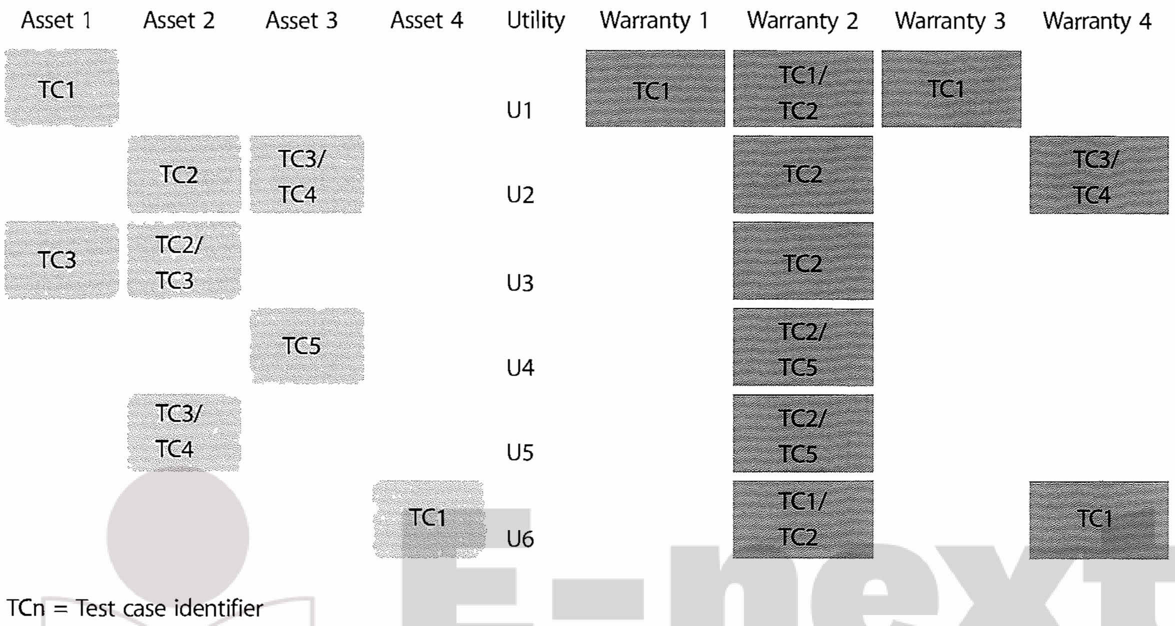
Service testing will include many non-functional tests. These tests can be conducted at several test levels to help build up confidence in the service release. They include:

- Usability testing
- Accessibility testing
- Process and procedure testing
- Knowledge transfer and competence testing
- Performance, capacity and resilience testing
- Volume, stress, load and scalability testing
- Availability testing
- Backup and recovery testing
- Coherency testing
- Compatibility testing
- Documentation testing
- Regulatory and compliance testing
- Security testing
- Logistics, deployability and migration testing
- Coexistence and compatibility testing
- Remediation, continuity and recovery testing
- Configuration, build and installability testing
- Operability and maintainability testing.

There are several types of testing from different perspectives, which are described below.

Service requirements and structure testing – service provider, users and customers

Validation of the service attributes against the contract, service package and service model includes evaluating the integration or 'fit' of the utilities across the core and supporting services and service assets to ensure there is complete coverage and no conflicts.



Asset 1	Asset 2	Asset 3	Asset 4	Utility	Warranty 1	Warranty 2	Warranty 3	Warranty 4
TC1				U1	TC1	TC1/ TC2	TC1	
	TC2	TC3/ TC4		U2		TC2		TC3/ TC4
TC3	TC2/ TC3			U3		TC2		
		TC5		U4		TC2/ TC5		
	TC3/ TC4			U5		TC2/ TC5		
			TC1	U6		TC1/ TC2		TC1

TCn = Test case identifier

Figure 4.31 Designing tests to cover range of service assets, utilities and warranties

Figure 4.31 shows a matrix of service utility to service warranty and the service assets that support each utility. This matrix is one that can be used to design the service tests to ensure that the service structure and test design coverage is appropriate. Service tests cases are designed to test the service requirements in terms of utility, capacity, resource utilization, finance and risks. For example approaches to testing the risk of service failure include performance, stress, usability and security testing.

Service level testing testing – service level managers, operations managers and customers

Validate that the service provider can deliver the service level requirements, e.g. testing the response and fix time, availability, product delivery times and support services.

The performance from a service asset should deliver the utility or service expected. This is not necessarily that the asset can deliver what it should be capable of in its own right. For example a car's factory specification may assert that it is capable of 150kph, but for most customers delivering 100kph will fully meet the requirement.

Warranty and assurance tests – fit for use testing

As discussed earlier in this section, the customers see the service delivered in terms of warranties against the utilities that add value to their assets in order to deliver the expected business support. For any service, the warranties are expressed in measurable terms that enable tests to be designed to establish that the warranty can be delivered (within the agreed degree of confidence). The degree of detail may vary considerably, but will always reflect the agreement established during Service Design. In all cases the warranty will be described, and should be measurable, in terms of the customer's business and the potential effects on it of success or failure of the service to meet that warranty.

The following tests are used to provide confidence that the warranties can be delivered, i.e. the service is fit for use:

- **Availability** is the most elementary aspect of assuring value to customers. It assures the customer that services will be available for use under agreed terms and conditions. Services are expected to be made available to designated users only within specified areas, locations and time schedules.

- **Capacity** is an aspect of service warranty that assures the customer that a service will support a specified level of business activity or demand at a specified level of service quality. Customers can make changes to their utilization of services while being assured that their business processes and systems will be adequately supported by the service. Capacity management is a critical aspect of Service Management because it has a direct impact on the availability of services. The capacity available to support services also has an impact of the level of service continuity committed or delivered. Effective management of service capacity can therefore have first-order and second-order effects on service warranty.
- **Continuity** is the level of assurance provided to customers that the service will continue to support the business through major failures or disruptive events. The service provider undertakes to maintain service assets that will provide a sufficient level of contingency and responsiveness. Specialized systems and processes will kick in to ensure that the service levels received by the customer's assets do not fall below a predefined level. Assurance is also provided that normal service levels will be restored within a predefined time limit to limit the overall impact of a failure or event. The effectiveness of service continuity is measured in terms of disturbance to the productive state of customer assets.
- **Security** assures that the utilization of services by customers will be secure. This means that customer assets within the scope of service delivery and support will not be exposed to certain security risks. Service providers undertake to implement general and service-level controls that will ensure that the value provided to customers is complete and not eroded by any avoidable costs and risks. Service security covers the following aspects of reducing risks:
    - Authorized and accountable usage of services as specified by customer
    - Protection of customers' assets from unauthorized or malicious access
    - Security zones between customer assets and service assets
    - Plays a supporting role to the other three aspects of service warranty
    - When effective has a positive impact on those aspects.

  Service security inherits all the general properties of the security of physical and human assets, as well as intangibles such as data, information, coordination and communication.

Usability – users and maintainers

Usability testing is likely to be of increasing importance as more services become widely used as a part of everyday life and ordinary business usage. Focusing on the intuitiveness of a service can significantly increase the efficiency and reduce the unit costs of both using and supporting a service.

User accessibility testing considers the restricted abilities of actual or potential users of a new or changed service and is commonly used for testing web services. Care must be taken to establish the types of likely users, e.g. hearing impaired users may be able to operate a PC-based service but would not be supported by a telephone-only-based service-desk support system. This testing might focus on usability for:

- Disabled users, e.g. visually or hearing impaired
- Sensory restricted users, e.g. colour-blind
- Users working in second language or based in a different culture.

Contract and regulation testing

Audits and tests are conducted to check that the criteria in contracts have been accepted before acceptance of the end-to-end service. Service providers may have a contractual requirement to comply with the requirements of ISO/IEC 20000 or other standards and they would need to ensure that the relevant clauses of the standard are met during implementation of a new or changed service and release.

Regulatory acceptance testing is required in some industries such as defence, financial services and pharmaceuticals.

Compliance testing

Testing is conducted to check compliance against internal regulations and existing commitments of the organization, e.g. fraud checks.

Service Management testing

The service models will dictate the approach to testing the integrated Service Management processes. ISO/IEC 20000 covers the minimum requirements for each process to be compliant with the standard and maintenance of the process interrelationships.

Examples of Service Management manageability tests are shown in Table 4.12.

Table 4.12 Examples of Service Management manageability tests

Service Management functions	Examples of design phase manageability checks	Examples of build phase manageability checks	Examples of deployment phase manageability checks	Examples of operating manageability checks	Examples of early life support and CSI manageability checks
Configuration Management	Are the designers aware of the corporate standards used for Configuration Management? How does the design meet organizational standards for acceptable configurations? Does the design support the concept of version control? Is the design created in a way that allows for the logical breakdown of the service into configuration items (CIs)?	Have the developers built the service, application and infrastructure to conform to the corporate standards that are used for Configuration Management? Does the service use only standard supporting systems and tools that are considered acceptable? Does the service include support for version, build, baseline and release control and management? Have the developers built in the chosen CI structure to the service, application and infrastructure?	Does the service deployment update the CMS at each stage of the rollout? Is the deployment team using an updated inventory to complete the plan and the deployment?	Can the operations team gain access to the CMS so that they can confirm the service they are managing is the correct version and configured correctly? Are the operating instructions under version and build control similar to those used for the application builds?	As the service is reviewed within the optimize phase, is the CMS used to assist with the review? Are Configuration Management personnel involved in the optimization process, including providing advice in the use of and updating the inventory?
Change Management	Does the Service Design cope with change? Do the designers understand the Change Management process used by the organization?	Have the service assets and components been built and tested against the corporate Change Management process? Has the emergency change process been tested? Is the impact assessment procedure for the CI type clearly defined and has it been tested?	Are the corporate Change Management process and standards used during deployment?	Is the operations team involved in the Change Management process; is it part of the sign-off and verification process? Does a member of the operations team attend the Change Management meetings?	As modifications are identified within this phase, does the team use the Change Management system to coordinate the changes? Does the optimization team understand the Change Management process?

Table 4.12 Examples of Service Management manageability tests (continued)

Service Management functions	Examples of design phase manageability checks	Examples of build phase manageability checks	Examples of deployment phase manageability checks	Examples of operating manageability checks	Examples of early life support and CSI manageability checks
Release and Deployment Management	Do the service designers understand the standards and tools used for releasing and deploying services? How will the design ensure that the new or changed service can be deployed into the environment in a simple and efficient way?	Has the service, application and infrastructure been built and tested in ways that ensure it can be released into the environment in a simple and efficient way?	Is the service being deployed in a manner that minimizes risks, such as a phased deployment? Has a remediation/ back-out option been included in the release package or process for the service and its constituent components?	Does the release and deployment process ensure that deployment information is available to the operations teams? Do the Service Operations teams have access to release and information even before the service or application is deployed into the live environment?	Do members of the CSI team understand the release process, and are they using this for planning the deployment of improvements? Is Release and Deployment Management involved in providing advice to the assessment process?
Security management	How does the design ensure that the service is designed with security in the forefront?	Is the build process following security best practice for this activity?	Can the service be deployed in a manner that meets organizational security standards and requirements?	Does the service support the ongoing and periodic checks that security management needs to complete while the service is in operational use?	
Incident management	Does the design facilitate simple creation of incidents when something goes wrong? Is the design compatible with the organizational incident management system? Does the design accommodate automatic logging and detection of incidents?	Is a simple creation-of-incidents process, for when something goes wrong, built into services and tested (e.g. notification from applications)? Has the compatibility with the organizational incident management system been tested?	Does the deployment use the incident management system for reporting issues and problems? Do the members of the deployment team have access to the incident management system so that they can record incidents and also view incidents that relate to the deployment?	Does the operations team have access to the incident management system and can it update information within this system? Does the operations team understand its responsibilities in dealing with incidents? Is the operations team provided with reports on how well it deals with incidents, and does it act on these?	Do members of the CSI team have access to the incident management system so that they can record incidents and also view incidents that may be addressed in optimization?

Table 4.12 Examples of Service Management manageability tests (continued)

Service Management functions	Examples of design phase manageability checks	Examples of build phase manageability checks	Examples of deployment phase manageability checks	Examples of operating manageability checks	Examples of early life support and CSI manageability checks
Problem management	How does the design facilitate the methods used for root cause analysis used within the organization?	Has the method of providing information to facilitate root cause analysis and problem management been tested?	Has a problem manager been appointed for this deployment and does the deployment team know who it is?	Does the operations team contribute to the problem management process, ideally by assisting with and facilitating root cause analysis? Does the operations team meet problem management staff regularly? Does the operations team see the weekly/ monthly problem management report?	Is the optimization process being provided with information by problem management to incorporate into the assessment process?
Capacity management	Are the designers aware of the approach to capacity management used within the organization? How to measure operations and performance? Is modelling being used to ensure that the design meets capacity needs?	Has the service been built and tested to ensure that it meets the capacity requirements? Has the capacity information provided by the service been tested and verified? Are stress and volume characteristics built into the services and constituent applications?	Is capacity management involved in the deployment process so that it can monitor the capacity of the resources involved in the deployment?	Is capacity management information being monitored and reported on as this service is used, and is this information provided to capacity management?	Is capacity management feeding information into the optimization process?
Availability management	Does the design address the availability requirements of the service? Has the service been designed to fit in with backup and recovery capabilities of the organization?	How has the service been built to address the availability requirements, and how has this been tested? What testing has been done to ensure that the service meets the backup and recovery capabilities of the organization? What happens when the service and underlying applications are under stress?	Is availability management monitoring the availability of the service, the applications being deployed and the rest of the technology infrastructure to ensure that the deployment is not affecting availability? How is the ability to back up and recover the service during deployment being dealt with?	How is the service's availability being measured, and is this information being fed back to the availability management function within the IT organization?	Does the assessment use the availability information to complete the proposal of modifications that are needed for the service? Is any improvement required in the service's ability to be backed up and recovered?

Table 4.12 Examples of Service Management manageability tests (continued)

Service Management functions	Examples of design phase manageability checks	Examples of build phase manageability checks	Examples of deployment phase manageability checks	Examples of operating manageability checks	Examples of early life support and CSI manageability checks
Service continuity management	How does the design meet the service continuity requirements of the organization? Will the design meet the needs of the business recovery process following a disaster?	Has the service been built to support the business recovery process following a disaster, and how has this been tested?	Will any changes be required to the business recovery process following a disaster if one should occur during or after the deployment of this service?	Is the business recovery process for the service tested regularly by operations?	What optimization is required in the business recovery process to meet the business needs?
Service level management	How does the design meet the SLA requirements of the organization?	Does the service meet the SLA and performance requirements, and has this been tested?	Is service level management aware of the deployment of this service? Does this service have an initial SLA for the deployment phase? Does the service affect the SLA requirements during deployment?	Is the SLA visible and understood by the operations team so that it appreciates how its running of the service affects the delivery of the SLA? Does operations see the weekly/monthly service level report?	Is service level management information available for inclusion in the optimization process?
Financial management	Does the design meet the financial requirements for this service? How does the design ensure that the final new or changed service will meet return of investment expectations?	Has the service been built to deliver financial information, and how is this being tested?	Is management accounting being done during the deployment so that the total cost of deployment can be included within the cost of ownership?	Does operations provide input into the financial information about the service? For example, if a service requires an operator to perform additional tasks at night, is this recorded?	Is financial information available to be included in the assessment process?

Operational tests – systems, services

There will be many operational tests depending on the type of service. Typical tests include:

- **Load and stress** – These tests establish if the new or changed service will perform to the required levels on the capacity likely to be available. The capacity elements may include any anticipated bottlenecks within the infrastructure that might be expected to restrict performance, including:
  - Load and throughput
  - Behaviour at the upper limits of system capability
  - Network bandwidth
  - Data storage
  - Processing power or live memory
  - Service desk resources – people and technology such as telephone lines and logging
  - Available software licences/concurrent seats
  - Support staff – both numbers and skills
  - Training facilities, classrooms, trainers, CBT licences etc.
  - Overnight batch processing timings, including backup tasks.
- **Security** – All services should be considered for their potential impact on relevant security concerns, and subsequently tested for their actual likely impact on security. Any service that has an anticipated security impact or exposes an anticipated security risk will have been assessed at design stage, and the requirement

for security involvement built into the service package. Organizations should make reference to and may wish to seek compliance with ISO 27000 where security is a significant concern to their services.

- **Recoverability** – Every significant change will have been assessed for the question 'If this change is made, will the Disaster Recovery (DR) plan need to be changed accordingly'. Notwithstanding that consideration earlier in the lifecycle, it is appropriate to test that the new or changed service is catered for within the existing (or amended with the changed) DR plan. Typically, concerns identified during testing should be addressed to the service continuity team and considered as active elements for future DR tests.

Regression testing

Regression testing means 'repeating a test already run successfully, and comparing the new results with the earlier valid results'. On each iteration of true regression testing, all existing, validated tests are run, and the new results are compared with the already-achieved standards. Regression testing ensures that a new or changed service does not introduce errors into aspects of the services or IT infrastructure that previously worked without error. Simple examples of the type of error that can be detected are software contention issues, hardware and network incompatibility. Regression testing also applies to other elements such as Service Management process testing and measurement. In reality it is the integrated concept of service testing – assessing whether the service will deliver the business benefit – that makes regression testing so very important in modern organizations, and will make it ever more important.

4.5.5 Process activities, methods and techniques

The testing process is shown schematically in Figure 4.32. The test activities are not undertaken in a sequence. Several activities may be done in parallel, e.g. test execution begins before all the test design is complete. The activities are described below.

1. Validation and test management

Test management includes the planning, control and reporting of activities through the test stages of Service Transition. These activities include:

- Planning the test resources
- Prioritizing and scheduling what is to be tested and when
- Management of incidents, problems, errors, non-conformances, risks and issues
- Checking that incoming known errors and their documentation are processed
- Monitoring progress and collating feedback from validation and test activities
- Management of incidents, problems, errors, non-conformances, risks and issues discovered during transition
- Consequential changes, to reduce errors going into production
- Capturing configuration baseline
- Test metrics collection, analysis, reporting and management.

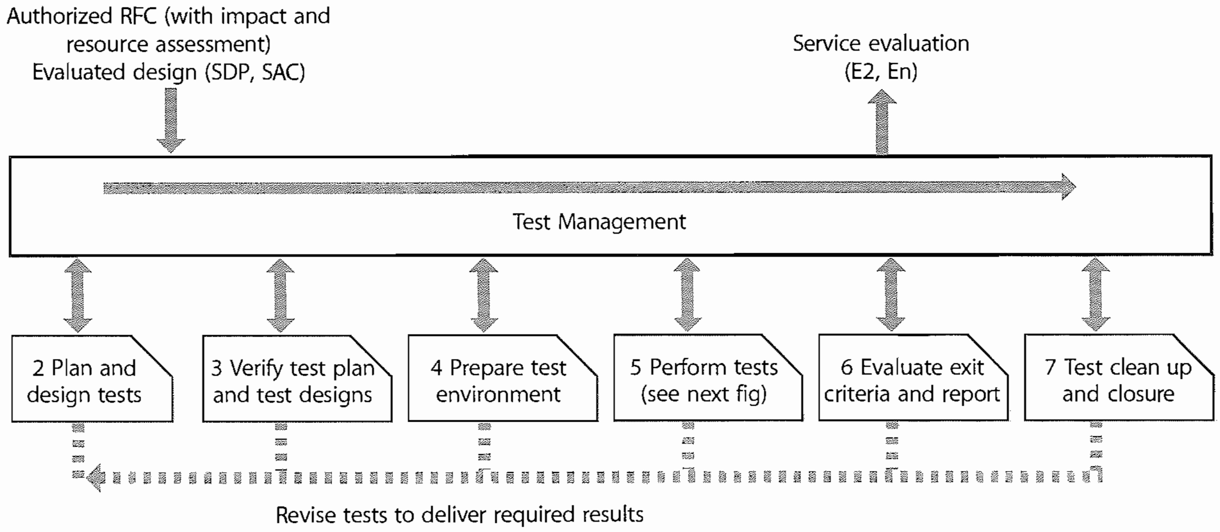

Authorized RFC (with impact and resource assessment)
Evaluated design (SDP, SAC)
Service evaluation (E2, En)
Test Management
2 Plan and design tests
3 Verify test plan and test designs
4 Prepare test environment
5 Perform tests (see next fig)
6 Evaluate exit criteria and report
7 Test clean up and closure
Revise tests to deliver required results

Figure 4.32 Example of a validation and testing process

Test management includes managing issues, mitigating risks and implementing changes identified from the testing activities as these can impose delays and create dependencies that need to be proactively managed.

Test metrics are used to measure the test process and manage and control the testing activities. They enable the test manager to determine the progress of testing, the earned value and the outstanding testing, and this helps the test manager to estimate when testing will be completed. Good metrics provide information for management decisions that are required for prioritization, scheduling and risk management. They also provide useful information for estimating and scheduling for future releases.

2. Plan and design test

Test planning and design activities start early in the service lifecycle and include:

- Resourcing
- Hardware, networking, staff numbers and skills etc. capacity
- Business/customer resources required, e.g. components or raw materials for production control services, cash for ATM services
- Supporting services including access, security, catering, communications
- Schedule of milestones, handover and delivery dates
- Agreed time for consideration of reports and other deliverables
- Point and time of delivery and acceptance
- Financial requirements – budgets and funding.

3. Verify test plan and test design

Verify the test plans and test design to ensure that:

- The test model delivers adequate and appropriate test coverage for the risk profile of the service
- The test model covers the key integration aspects and interfaces, e.g. at the SPIs
- That the test scripts are accurate and complete.

4. Prepare test environment

Prepare the test environment by using the services of the build and test environment resource and also use the release and deployment processes to prepare the test environment where possible; see paragraph 4.4.5.2. Capture a configuration baseline of the initial test environment.

5. Perform tests

Carry out the tests using manual or automated techniques and procedures. Testers must record their findings during the tests. If a test fails, the reasons for failure must be fully documented. Testing should continue according to the test plans and scripts, if at all possible. When part of a test fails, the incident or issues should be resolved or documented (e.g. as a known error) and the appropriate re-tests should be performed by the same tester.

An example of the test execution activities is shown in Figure 4.33. The deliverables from testing are:

- Actual results showing proof of testing with cross-references to the test model, test cycles and conditions
- Problems, errors, issues, non-conformances and risks remaining to be resolved
- Resolved problems/known errors and related changes
- Sign-off.

6. Evaluate exit criteria and report

The actual results are compared to the expected results. The results may be interpreted in terms of pass/fail; risk to the business/service provider; or if there is a change in a projected value, e.g. higher cost to deliver intended benefits.

To produce the report, gather the test metrics and summarize the results of the tests. Examples of exit criteria are:

- The service, with its underlying applications and technology infrastructure, enables the business users to perform all aspects of function as defined.
- The service meets the quality requirements.
- Configuration baselines are captured into the CMS.

7. Test clean up and closure

Ensure that the test environments are cleaned up or initialized. Review the testing approach and identify improvements to input to design/build, buy/build decision parameters and future testing policy/procedures.

4.5.6 Trigger, input and outputs, and inter-process interfaces

4.5.6.1 Trigger

The trigger for testing is a scheduled activity on a release plan, test plan or quality assurance plan.

4.5.6.2 Inputs

The key inputs to the process are:

- **The service package** – This comprises a core service package and re-usable components, many of which themselves are services, e.g. supporting service. It defines the service's utilities and warranties that are

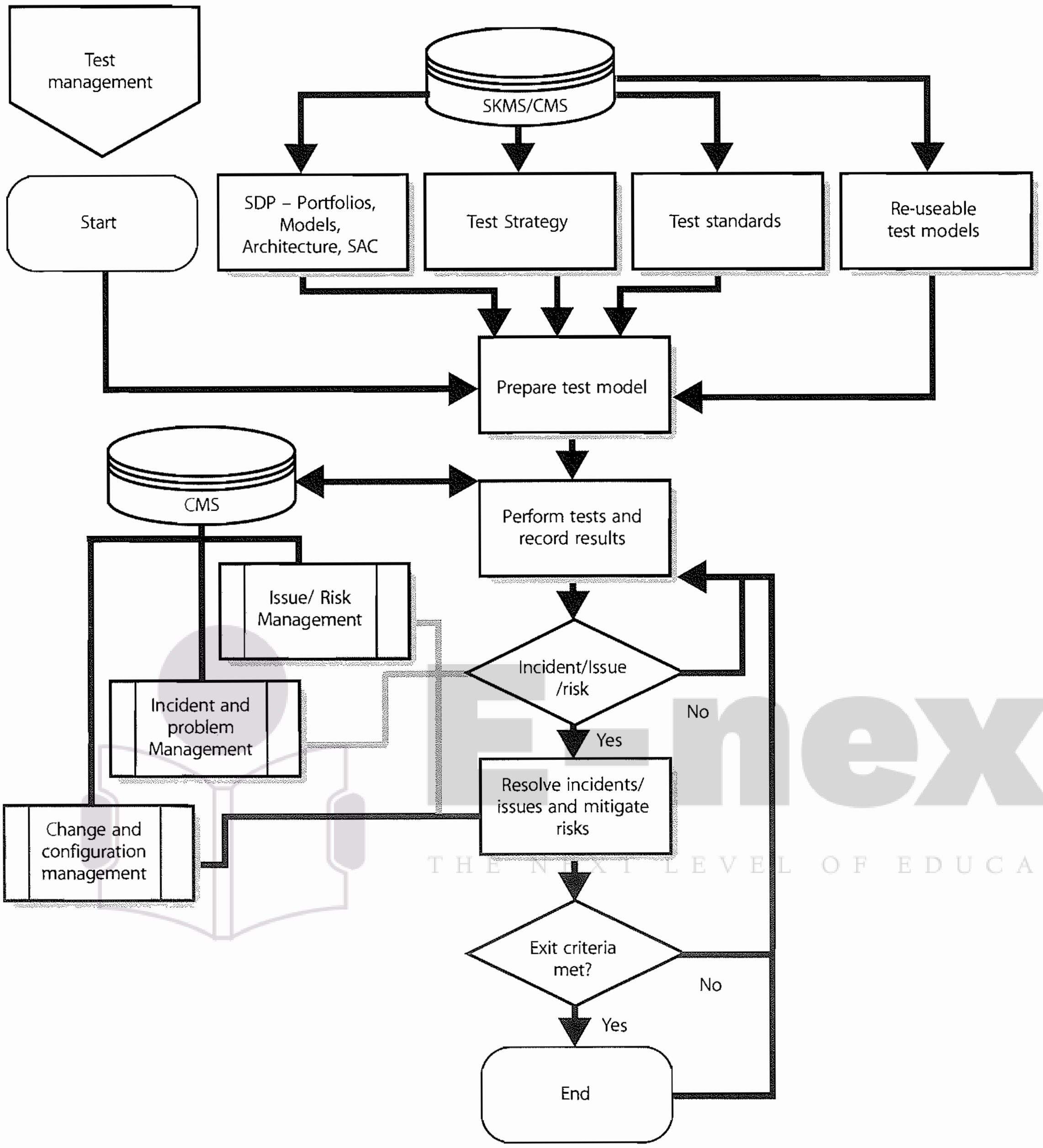

Test management
SKMS/CMS
Start
SDP – Portfolios, Models, Architecture, SAC
Test Strategy
Test standards
Re-useable test models
Prepare test model
CMS
Perform tests and record results
Issue/ Risk Management
Incident/Issue /risk
No
Yes
Incident and problem Management
Resolve incidents/ issues and mitigate risks
Change and configuration management
Exit criteria met?
No
Yes
End

Figure 4.33 Example of perform test activities

delivered through the correct functioning of the particular set of identified service assets. It maps the demand patterns for service and user profiles to SLPs.

- **SLP** – One or more SLPs that provided a definitive level of utility or warranty from the perspective of outcomes, assets, patterns of business activity of customers (PBA).
- **Service provider interface definitions** – These define the interfaces to be tested at the boundaries of the service being delivered, e.g. process interfaces, organizational interfaces.
- **The Service Design package** – This defines the agreed requirements of the service, expressed in terms of the service model and Service Operations plan. It includes:
  - Operation models (including support resources, escalation procedures and critical situation handling procedures)
  - Capacity/resource model and plans – combined with performance and availability aspects
  - Financial/economic/cost models (with TCO, TCU)
  - Service Management model (e.g. integrated process model as in ISO/IEC 20000)
  - Design and interface specifications.

- **Release and deployment plans** – These define the order that release units will be deployed, built and installed.
- **Acceptance Criteria** – These exist at all levels at which testing and acceptance are foreseen.
- **RFCs** – These instigate required changes to the environment within which the service functions or will function.

4.5.6.3 Outputs

The direct output from testing is the report delivered to service evaluation (see section 4.6). This sets out:

- Configuration baseline of the testing environment
- Testing carried out (including options chosen and constraints encountered)
- Results from those tests
- Analysis of the results, e.g. comparison of actual results with expected results, risks identified during testing activities.

After the service has been in use for a reasonable time there should be sufficient data to perform an evaluation of the actual vs predicted service capability and performance. If the evaluation is successful, an evaluation report is sent to Change Management with a recommendation to promote the service release out of early life support and into normal operation.

Other outputs include:

- Updated data, information and knowledge to be added to the service knowledge management system, e.g. errors and workarounds, testing techniques, analysis methods
- Test incidents, problems and error records
- Improvement ideas for Continual Service Improvement to address potential improvements in any area that impacts on testing:
  - To the testing process itself
  - To the nature and documentation of the Service Design outputs
- Third party relationships, suppliers of equipment or services, partners (co-suppliers to end customers), users and customers or other stakeholders.

4.5.6.4 Interfaces to other lifecycle stages

Testing supports all of the release and deployment steps within Service Transition.

Although this chapter focuses on the application of testing within the Service Transition phase, the test strategy will ensure that the testing process works with all stages of the lifecycle:

- Working with Service Design to ensure that designs are inherently testable and providing positive support in achieving this; examples range from including self-monitoring within hardware and software, through the re-use of previously tested and known service elements through to ensuring rights of access to third party suppliers to carry out inspection and observation on delivered service elements easily
- Working closely with CSI to feed failure information and improvement ideas resulting from testing exercises
- Service Operation will use maintenance tests to ensure the continued efficacy of services; these tests will require maintenance to cope with innovation and change in environmental circumstances
- Service Strategy should accommodate testing in terms of adequate funding, resource, profile etc.

4.5.7 Information management

The nature of IT Service Management is repetitive, and this ability to benefit from re-use is recognized in the suggested use of transition models. Testing benefits greatly from re-use and to this end it is sensible to create and maintain a library of relevant tests and an updated and maintained data set for applying and performing tests. The test management group within an organization should take responsibility for creating, cataloguing and maintaining test scripts, test cases and test data that can be re-used.

Similarly, the use of automated testing tools (Computer Aided Software Testing – CAST) is becoming ever more central to effective testing in complex software environments. Equivalently standard and automated hardware testing approaches are fast and effective.

Test data

However well a test has been designed, it relies on the relevance of the data used to run it. This clearly applies strongly to software testing, but equivalent concerns relate to the environments within which hardware, documentation etc. is tested. Testing electrical equipment in a protected environment, with smoothed power supply and dust, temperature and humidity control will not be a valuable test if the equipment will be used in a normal office.

Test environments

Test environments must be actively maintained and protected. For any significant change to a service, the question should be asked (as for continued relevance of the continuity and capacity plans, should the change be accepted and implemented): 'If this change goes ahead, will there need to be a consequential impact to the test data?' If so, it may involve updating test data as part of the change, and the dependency of a service, or service element, on test data or test environment will be evident from the SKMS, via records and relationships held within the CMS. Outcomes from this question include:

- Consequential updating of the test data
- A new separate set of data or new test environment, since the original is still required for other services
- Redundancy of the test data or environment – since the change will allow testing within another existing test environment, with or without modification to that data/environment (this may in fact be the justification behind a perfective change – to reduce testing costs)
- Acceptance that a lower level of testing will be accepted since the test data/environment cannot be updated to deliver equivalent test coverage for the changed service.

Maintenance of test data should be an active exercise and should address relevant issues including:

- Separation from any live data, and steps to ensure that it cannot be mistaken for live data when being used, and vice versa (there are many real-life examples of live data being copied and used as test data and being the basis for business decisions)
- Data protection regulations – when live data is used to generate a test database, if information can be traced to individuals it may well be covered by data protection legislation that, for example, may forbid its transportation between countries
- Backup of test data, and restoration to a known baseline for enabling repeatable testing; this also applies to initiation conditions for hardware tests that should be baselined.

An established test database can also be used as a safe and realistic training environment for a service.

4.5.8 Key performance indicators and metrics

4.5.8.1 Primary (of value to the business/customers)

The business will judge testing performance as a component of the Service Design and transition stages of the service lifecycle. Specifically, the effectiveness of testing in delivering to the business can be judged through:

- Early validation that the service will deliver the predicted value that enables early correction
- Reduction in the impact of incidents and errors in live that are attributable to newly transitioned services
- More effective use of resource and involvement from the customer/business
- Reduced delays in testing that impact the business
- Increased mutual understanding of the new or changed service
- Clear understanding of roles and responsibilities associated with the new or changed service between the customers, users and service provider
- Cost and resources required from user and customer involvement (e.g. user acceptance testing).

The business will also be concerned with the economy of the testing process – in terms of:

- Test planning, preparation, execution rates
- Incident, problem, event rates
- Issue and risk rate
- Problem resolution rate
- Resolution effectiveness rate
- Stage containment – analysis by service lifecycle stage
- Repair effort percentage
- Problems and changes by service asset or CI type
- Late changes by service lifecycle stage
- Inspection effectiveness percentage
- Residual risk percentage
- Inspection and testing return on investment (ROI)
- Cost of unplanned and unbudgeted overtime to the business
- Cost of fixing errors in live operation compared to fixing errors early in the lifecycle (e.g. the costs can be £10 to fix an error in Service Design and £10,000 to fix the error if it reaches production)
- Operational cost improvements associated with reducing errors in new or changed services.

4.5.8.2 Secondary (internal)

The testing function and process itself must strive to be effective and efficient, and so measures of its effectiveness and costs need to be taken. These include:

- Effort and cost to set up a testing environment
- Effort required to find defects – i.e. number of defects (by significance, type, category etc.) compared with testing resource applied
- Reduction of repeat errors – feedback from testing ensures that corrective action within design and transition (through CSI) prevents mistakes from being repeated in subsequent releases or services

- Reduced error/defect rate in later testing stages or production
- Re-use of testing data
- Percentage incidents linked to errors detected during testing and released into live
- Percentage errors at each lifecycle stage
- Number and percentage of errors that could have been discovered in testing
- Testing incidents found as percentage of incidents occurring in live operations
- Percentage of faults found in earlier assessment stages – since remedial costs accelerate steeply for correction in later stages of transition
- Number of known errors documented in earlier testing phases.

Testing is about measuring the ability of a service to perform as required in a simulated (or occasionally the actual) environment, and so to that extent is focused on measurement. Care must be taken to try and separate out the measures that actually relate to the testing process from the number of errors introduced into services and systems. Careless measurement can appear to improve testing effectiveness although the development practices are worse – it is simply easier to find defects when there are lots of them. The point here is that testing is actually a stage of the design, build, release and deployment processes and the important measure is the overall one – about delivering services that deliver benefits and fail less often.

4.5.9 Challenges, critical success factors and risks

Still the most frequent challenges to effective testing are based on lack of respect and understanding for the role of testing. Traditionally testing has been starved of funding, and this results in:

- Inability to maintain test environment and test data that matches the live environment
- Insufficient staff, skills and testing tools to deliver adequate testing coverage
- Projects overrunning and allocated testing time frames being squeezed to restore project go-live dates but at the cost of quality
- Developing standard performance measures and measurement methods across projects and suppliers
- Projects and suppliers estimating delivery dates inaccurately and causing delays in scheduling Service Transition activities.

Critical success factors include:

- Understanding the different stakeholder perspectives that underpin effective risk management for the change impact assessment and test activities
- Building a thorough understanding of risks that have impacted or may impact successful Service Transition of services and releases
- Encouraging a risk management culture where people share information and take a pragmatic and measured approach to risk.
- Quality is built into every stage of the service lifecycle using a structured framework such as the V-model
- Issues are identified early in the service lifecycle
- Testing provides evidence that the service assets and configurations have been built and implemented correctly in addition to the service delivering what the customer needs
- Re-usable test models are developed that can be used for regression testing in future releases

Risks to successful Service Validation and Testing include:

- Unclear expectations/objectives
- Lack of understanding of the risks means that testing is not targeted at critical elements that need to be well controlled and therefore tested
- Resource shortages (e.g. users, support staff) introduce delays and have an impact on other Service Transitions.

4.6 EVALUATION

Evaluation is a generic process that considers whether the performance of something is acceptable, value for money etc. – and whether it will be proceeded with, accepted into use, paid for, etc.

4.6.1 Purpose, goal and objective

The purpose of evaluation is to provide a consistent and standardized means of determining the performance of a service change in the context of existing and proposed services and IT infrastructure. The actual performance of a change is assessed against its predicted performance and any deviations between the two are understood and managed.

The goal of evaluation is to set stakeholder expectations correctly and provide effective and accurate information to Change Management to make sure changes that adversely affect service capability and introduce risk are not transitioned unchecked.

The objective is to:

- Evaluate the intended effects of a service change and as much of the unintended effects as is reasonably practical given capacity, resource and organizational constraints
- Provide good quality outputs from the evaluation process so that Change Management can expedite an effective decision about whether a service change is to be approved or not.

4.6.2 Scope

Specifically in this section we consider the evaluation of new or changed services defined by Service Design, during deployment and before final transition to service operations. The importance of evaluating the actual performance of any service change against its anticipated performance is an important source of information to service providers to help ensure that expectations set are realistic and to identify that if there are any reasons that production performance does not meet what was expected.

4.6.3 Value to business

Evaluation is, by its very nature, concerned with value. Specifically effective evaluation will establish the use made of resources in terms of delivered benefit and this information will allow a more accurate focus on value in future service development and Change Management. There is a great deal of intelligence that Continual Service Improvement can take from evaluation to analyse future improvements to the process of change and the predictions and measurement of service change performance.

4.6.4 Policies, principles and basic concepts

Policies

The following policies apply to the evaluation process:

- Service Designs or service changes will be evaluated before being transitioned.
- Any deviation between predicted and actual performance will be managed by the customer or customer representative by accepting the change even though actual performance is different to what was predicted; rejecting the change; or requiring a new change to be implemented with revised predicted performance agreed in advance. No other outcomes of evaluation are allowed.
- An evaluation shall not be performed without a customer engagement package.

Principles

The following principles shall guide the execution evaluation process:

- As far as is reasonably practical, the unintended as well as the intended effects of a change need to be identified and their consequences understood and considered.
- A service change will be fairly, consistently, openly and, wherever possible, objectively evaluated.

Basic concepts

The evaluation process uses the Plan–Do–Check–Act (PDCA) model to ensure consistency across all evaluations.

4.6.5 Process activities, methods and techniques

4.6.5.1 Service evaluation terms

The key terms shown in Table 4.13 apply to the service evaluation process.

Table 4.13 Key terms that apply to the service evaluation process

Term	Function/Means
Service change	A change to an existing service or the introduction of a new service; the service change arrives into service evaluation and qualification in the form of a Request for Change (RFC) from Change Management
Service Design package	Defines the service and provides a plan of service changes for the next period (e.g. the next 12 months). Of particular interest to service evaluation is the Acceptance Criteria and the predicted performance of a service with respect to a service change
Performance	The utilities and warranties of a service
Performance model	A representation of the performance of a service
Predicted performance	The expected performance of a service following a service change
Actual performance	The performance achieved following a service change
Deviations report	The difference between predicted and actual performance
Risk	A function of the likelihood and negative impact of a service not performing as expected
Countermeasures	The mitigation that is implemented to reduce risk
Test plan and results	The test plan is a response to an impact assessment of the proposed service change. Typically the plan will specify how the change will be tested; what records will result from testing and where they will be stored; who will approve the change; and how it will be ensured that the change and the service(s) it affects will remain stable over time. The test plan may include a qualification plan and a validation plan if the change affects a regulated environment. The results represent the actual performance following implementation of the change
Residual risk	The remaining risk after countermeasures have been deployed
Service capability	The ability of a service to perform as required
Capacity	An organization's ability to maintain service capability under any predefined circumstances
Constraint	Limits on an organization's capacity
Resource	The normal requirements of an organization to maintain service capability
Evaluation plan	The outcome of the evaluation planning exercise
Evaluation report	A report generated by the evaluation function, which is passed to Change Management and which comprises: ▨ A risk profile ▨ A deviations report ▨ A recommendation ▨ A qualification statement.

4.6.5.2 Evaluation process

Figure 4.34 shows the evaluation process with inputs and outputs.

4.6.5.3 Evaluation plan

Evaluation of a change should be carried out from a number of different perspectives to ensure any unintended effects of a change are understood as well as the intended effects.

Generally speaking we would expect the intended effects of a change to be beneficial. The unintended effects are harder to predict, often not seen even after the service change is implemented, and frequently ignored. Additionally, they will not always be beneficial, for example in terms of impact on other services, impact on customers and users of the service, and network overloading.

Intended effects of a change should match the Acceptance Criteria. Unintended effects are often not seen until pilot stage or even once in production; they are difficult to measure and very often not beneficial to the business.

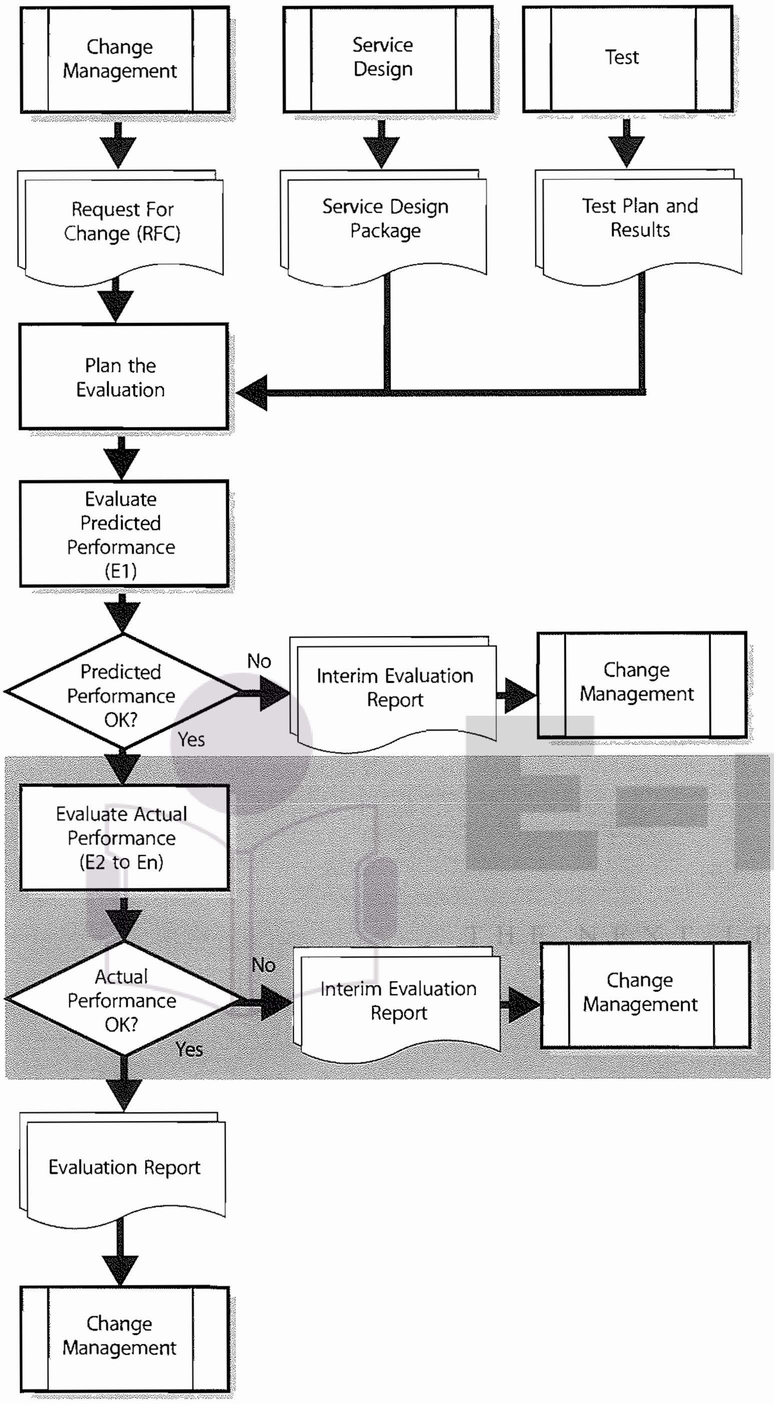


Figure 4.34 Evaluation process

4.6.5.4 Understanding the intended effect of a change

The details of the service change, customer requirements and Service Design package should be carefully analysed to understand fully the purpose of the change and the expected benefit from implementing it. Examples might include: reduce cost of running the service; increase service performance; reduce resources required to operate the service; or improve service capability.

The change documentation should make clear what the intended effect of the change will be and specific measures that should be used to determine effectiveness of that change. If they are in any way unclear or ambiguous the evaluation should cease and a recommendation not to proceed should be forwarded to Change Management.

Even some deliberately designed changes may be detrimental to some elements of the service. For example, the introduction of SOX-compliant procedures, which, while delivering the benefit of legal compliance, introduce extra work steps and costs.

4.6.5.5 Understanding the unintended effect of a change

In addition to the expected effects on the service and broader organization there are likely to be additional effects which were not expected or planned for. These effects must also be surfaced and considered if the full impact of a service change is to be understood. One of the most effective ways of identifying such effects is by discussion with all stakeholders. Not just customers, but also users of the service, those who maintain it, those who fund it etc. Care should be taken in presenting the details of the change to ensure stakeholders fully understand the implications and can therefore provide accurate feedback.

4.6.5.6 Factors for considering the effect of a service change

Table 4.14 shows the factors to be included when considering the effect of a service change.

4.6.5.7 Evaluation of predicted performance

Using customer requirements (including Acceptance Criteria), the predicted performance and the performance model, a risk assessment is carried out. If the risk assessment suggests that predicted performance may create unacceptable risks from the change or not meet the Acceptance Criteria, an interim evaluation report is sent to alert Change Management.

The interim evaluation report includes the outcome of the risk assessment and/or the outcome of the predicted performance versus Acceptance Criteria, together with a recommendation to reject the service change in its current form.

Evaluation activities cease at this point pending a decision from Change Management.

4.6.5.8 Evaluation of actual performance

Once the service change has been implemented a report on actual performance is received from operations. Using customer requirements (including Acceptance Criteria), the actual performance and the performance model, a risk assessment is carried out. Again if the risk assessment suggests that actual performance is creating unacceptable risks, an interim evaluation report is sent to Change Management.

The interim evaluation report includes the outcome of the risk assessment and/or the outcome of the actual performance versus Acceptance Criteria, together with a recommendation to remediate the service change.

Evaluation activities cease at this point pending a decision from Change Management.

4.6.5.9 Risk management

There are two steps in risk management: risk assessment and mitigation. Risk assessment is concerned with analysing threats and weaknesses that have been or would be introduced as a result of a service change.

Table 4.14 Factors for considering the effects of a service change

Factor	Evaluation of Service Design
S – Service provider capability	The ability of a service provider or service unit to perform as required.
T – Tolerance	The ability or capacity of a service to absorb the service change or release.
O – Organizational setting	The ability of an organization to accept the proposed change. For example, is appropriate access available for the implementation team? Have all existing services that would be affected by the change been updated to ensure smooth transition?
R – Resources	The availability of appropriately skilled and knowledgeable people, sufficient finances, infrastructure, applications and other resources necessary to run the service following transition.
M – Modelling and measurement	The extent to which the predictions of behaviour generated from the model match the actual behaviour of the new or changed service.
P – People	The people within a system and the effect of change on them.
U – Use	Will the service be fit for use? The ability to deliver the warranties, e.g. continuously available, is there enough capacity, will it be secure enough?
P – Purpose	Will the new or changed service be fit for purpose? Can the required performance be supported? Will the constraints be removed as planned?

A risk occurs when a threat can exploit a weakness. The likelihood of threats exploiting a weakness and the impact if they do, are the fundamental factors in determining risk.

The risk management formula is simple but very powerful:

Risk = Likelihood x Impact

Obviously, the introduction of new threats and weaknesses increases the likelihood of a threat exploiting a weakness. Placing greater dependence on a service or component increases the impact if an existing threat exploits an existing weakness within the service. These are just a couple of examples of how risk may increase as a result of a service change.

It is a clear requirement that a proposed service change must assess the existing risks within a service and the predicted risks following implementation of the change.

If the risk level has increased then the second stage of risk management is used to mitigate the risk. In the examples given above mitigation may include steps to eliminate a threat or weakness and using disaster recovery and backup techniques to increase the resilience of a service on which the organization has become more dependent.

Following mitigation the risk level is re-assessed and compared with the original. This second assessment and any subsequent assessments are in effect determining residual risk – the risk that remains after mitigation. Assessment of residual risk and associated mitigation continues to cycle until risk is managed down to an acceptable level.

The guiding principle here is that either the initial risk assessment or any residual risk level is equal to or less than the original risk prior to the service change. If this is not the case then evaluation will recommend rejection of proposed service change, or back out of an implemented service change.

The approach to risk representation recommended here takes a fundamentally different approach. Building on the work of Drake (2005a, 2005b) this approach recognizes that risks almost always grow exponentially over time if left unmanaged, and that a risk that will not cause a loss probably is not worth worrying about too much.

It is therefore proposed that a stronger risk representation is as shown in Figure 4.35. Principally, this representation is intended to promote debate and agreement by stakeholders: is the risk positioned correctly in terms of time and potential or actual loss; could mitigation have been deployed later (e.g. more economically); should it have been deployed earlier (e.g. better protection); etc.

Deviations – predicted vs actual performance

Once the service change passes the evaluation of predicted performance and actual performance, essentially as standalone evaluations, a comparison of the two is carried out. To have reached this point it will have been determined that predicted performance and actual performance are acceptable, and that there are no unacceptable risks. The output of this activity is a deviations report. For each factor in Table 4.14 the report states the extent of any deviation between predicted and actual performance.

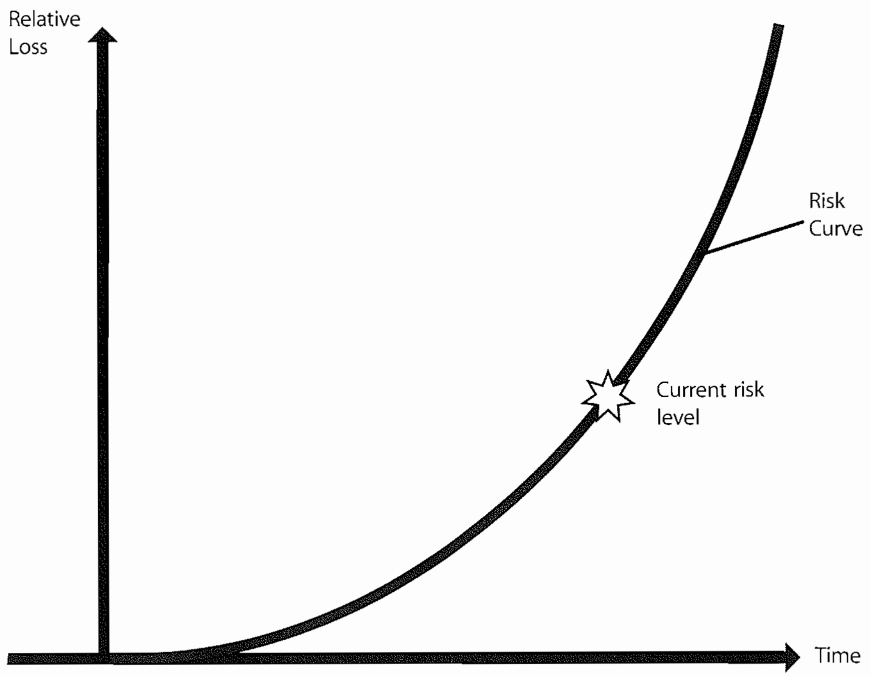


Figure 4.35 Example risk profile

Test plan and results

The testing function provides the means for determining the actual performance of the service following implementation of a service change. Test provides the service evaluation function with the test plan and a report on the results of any testing. The actual results are also made available to service evaluation. These are evaluated and used as described in paragraph 4.6.5.8.

In some circumstances it is necessary to provide a statement of qualification and/or validation status following a change. This takes place in regulated environments such as pharmaceuticals and defence.

The context for these activities is shown in Figure 4.36.

The inputs to these activities are the qualification plan and results and/or validation plan and results. The evaluation process ensures that the results meet the requirements of the plans. A qualification and/or validation statement is provided as output.

4.6.6 Evaluation report

The evaluation report contains the following sections.

Risk profile

A representation of the residual risk left after a change has been implemented and after countermeasures have been applied.

Deviations report

The difference between predicted and actual performance following the implementation of a change.

A qualification statement (if appropriate)

Following review of qualification test results and the qualification plan, a statement of whether or not the change has left the service in a state whereby it could not be qualified.

A validation statement (if appropriate)

Following review of validation test results and the validation plan, a statement of whether or not the change has left the service in a state whereby it could not be validated.

A recommendation

Based on the other factors within the evaluation report, a recommendation to Change Management to accept or reject the change:

- Review and close transition
- Knowledge Management.

4.6.7 Triggers, inputs and outputs and inter-process interfaces

Triggers:

- Request for Evaluation from Service Transition manager or Change Management
- Activity on Project Plan.

Inputs:

- Service package
- SDP and SAC
- Test results and report.

Outputs:

- Evaluation report for Change Management.

4.6.8 Information management

- Service Portfolio
- Service package
- SDP, SAC
- Test results and report
- Evaluation report.

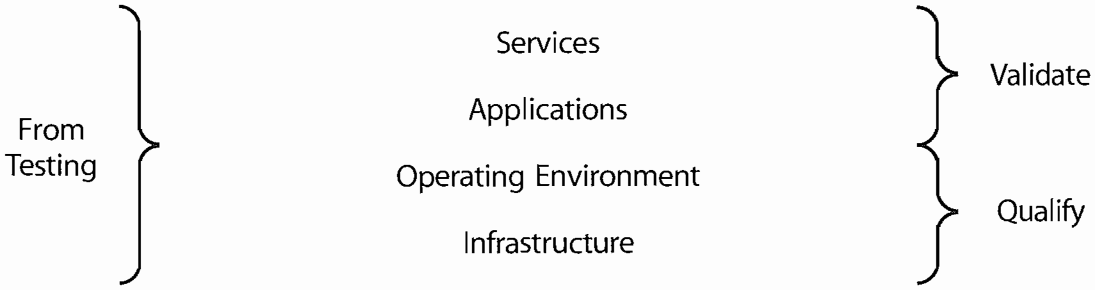


Figure 4.36 Context for qualification and validation activities

4.6.9 Key performance indicators and metrics

The customer/business KPIs are:

- Variance from service performance required by customers (minimal and reducing)
- Number of incidents against the service (low and reducing).

The internal KPIs include:

- Number of failed designs that have been transitioned (zero)
- Cycle time to perform an evaluation (low and reducing).

4.6.9.1 Challenges

Challenges include:

- Developing standard performance measures and measurement methods across projects and suppliers
- Projects and suppliers estimating delivery dates inaccurately and causing delays in scheduling evaluation activities
- Understanding the different stakeholder perspectives that underpin effective risk management for the evaluation activities
- Understanding, and being able to assess, the balance between managing risk and taking risks as it affects the overall strategy of the organization and service delivery
- Measuring and demonstrating less variation in predictions during and after transition
- Taking a pragmatic and measured approach to risk
- Communicating the organization's attitude to risk and approach to risk management effectively during risk evaluation
- Building a thorough understanding of risks that have impacted or may impact successful Service Transition of services and releases
- Encouraging a risk management culture where people share information.

4.7 KNOWLEDGE MANAGEMENT

The ability to deliver a quality service or process rests to a significant extent on the ability of those involved to respond to circumstances – and that in turn rests heavily on their understanding of the situation, the options and the consequences and benefits, i.e. their knowledge of the situation they are, or may find themselves, in.

That knowledge within the Service Transition domain might include:

- Identity of stakeholders
- Acceptable risk levels and performance expectations
- Available resource and timescales.

The quality and relevance of the knowledge rests in turn on the accessibility, quality and continued relevance of the underpinning data and information available to service staff.

4.7.1 Purpose, goal and objective

The purpose of Knowledge Management is to ensure that the right information is delivered to the appropriate place or competent person at the right time to enable informed decision.

The goal of Knowledge Management is to enable organizations to improve the quality of management decision making by ensuring that reliable and secure information and data is available throughout the service lifecycle.

The objectives of Knowledge Management include:

- Enabling the service provider to be more efficient and improve quality of service, increase satisfaction and reduce the cost of service
- Ensuring staff have a clear and common understanding of the value that their services provide to customers and the ways in which benefits are realized from the use of those services
- Ensuring that, at a given time and location, service provider staff have adequate information on:
  - Who is currently using their services
  - The current states of consumption
  - Service delivery constraints
  - Difficulties faced by the customer in fully realizing the benefits expected from the service.

4.7.2 Scope

Knowledge Management is a whole lifecycle-wide process in that it is relevant to all lifecycle sectors and hence is referenced throughout ITIL from the perspective of each publication. It is dealt with to some degree within other ITIL publications but this chapter sets out the basic concept, from a Service Transition focus.

4.7.2.1 Inclusions

Knowledge Management includes oversight of the management of knowledge, the information and data from which that knowledge derives.

4.7.2.2 Exclusions

Detailed attention to the capturing, maintenance and use of asset and configuration data is set out in Section 4.2.

4.7.3 Value to business

Knowledge Management is especially significant within Service Transition since relevant and appropriate knowledge is one of the key service elements being transitioned. Examples where successful transition rests on appropriate Knowledge Management include:

- User, service desk, support staff and supplier understanding of the new or changed service, including knowledge of errors signed off before deployment, to facilitate their roles within that service
- Awareness of the use of the service, and the discontinuation of previous versions
- Establishment of the acceptable risk and confidence levels associated with the transition, e.g. measuring, understanding and acting correctly on results of testing and other assurance results.

Effective Knowledge Management is a powerful asset for people in all roles across all stages of the service lifecycle. It is an excellent method for individuals and teams to share data, information and knowledge about all facets of an IT service. The creation of a single system for Knowledge Management is recommended.

Specific application to Service Transition domain can be illustrated through considering the following examples:

- Blurring of the concept of intellectual property and information when engaged in sourcing and partnering, therefore new approaches to controlling 'knowledge' must be addressed and managed during Service Transition
- Knowledge transfer often being a crucial factor in facilitating effective transition of new or changed services and essential to operational readiness
- Training of users, support staff, suppliers and other stakeholders in new or changed services
- Recording of errors, faults, workarounds etc. detected and documented during the Service Transition phase
- Capturing of implementation and testing information
- Re-using previously developed and quality assured testing, training and documentation
- Compliance with legislative requirements, e.g. SOX, and conformance to standards such as ISO 9000 and ISO/IEC 20000
- Assisting decisions on whether to accept or proceed with items and services by delivering all available relevant information (and omitting unnecessary and confusing information) to key decision makers.

4.7.4 Policies, principles and basic concepts

4.7.4.1 The Data-to-Information-to-Knowledge-to-Wisdom structure

Knowledge Management is typically displayed within the Data-to-Information-to-Knowledge-to-Wisdom (DIKW) structure. The use of these terms is set out below.

Data is a set of discrete facts about events. Most organizations capture significant amounts of data in highly structured databases such as Service Management and Configuration Management tools/systems and databases.

The key Knowledge Management activities around data are the ability to:

- Capture accurate data
- Analyse, synthesize, and then transform the data into information
- Identify relevant data and concentrate resources on its capture.

Information comes from providing context to data. Information is typically stored in semi-structured content such as documents, e-mail, and multimedia.

The key Knowledge Management activity around information is managing the content in a way that makes it easy to capture, query, find, re-use and learn from experiences so that mistakes are not repeated and work is not duplicated.

Knowledge is composed of the tacit experiences, ideas, insights, values and judgements of individuals. People gain knowledge both from their own and from their peers' expertise, as well as from the analysis of information (and data). Through the synthesis of these elements, new knowledge is created.

Knowledge is dynamic and context based. Knowledge puts information into an 'ease of use' form, which can facilitate decision making. In Service Transition this knowledge is not solely based on the transition in progress, but is gathered from experience of previous transitions, awareness of recent and anticipated changes and other areas that experienced staff will have been unconsciously collecting for some time.

Wisdom gives the ultimate discernment of the material and having the application and contextual awareness to provide a strong common sense judgement.

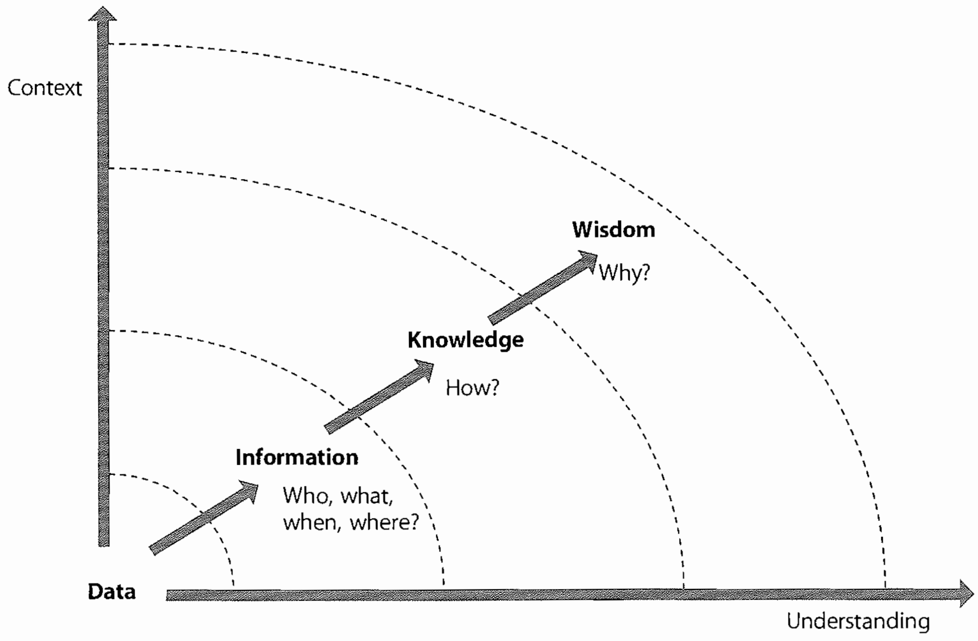


Figure 4.37 The flow from data to wisdom

This is shown in Figure 4.37.

4.7.4.2 The service knowledge management system

Specifically within IT Service Management, Knowledge Management will be focused within the Service Knowledge Management System (SKMS) concerned, as its name implies, with knowledge. Underpinning this knowledge will be a considerable quantity of data, which will be held in a central logical repository or Configuration Management System (CMS) and Configuration Management Database (CMDB). However, clearly the SKMS is a broader concept that covers a much wider base of knowledge, for example:

- The experience of staff
- Records of peripheral matters, e.g. weather, user numbers and behaviour, organization's performance figures
- Suppliers' and partners' requirements, abilities and expectations
- Typical and anticipated user skill levels.

Figure 4.38 is a very simplified illustration of the relationship of the three levels, with data being gathered within the CMDB, and feeding through the CMS into the SKMS and supporting the informed decision making process.

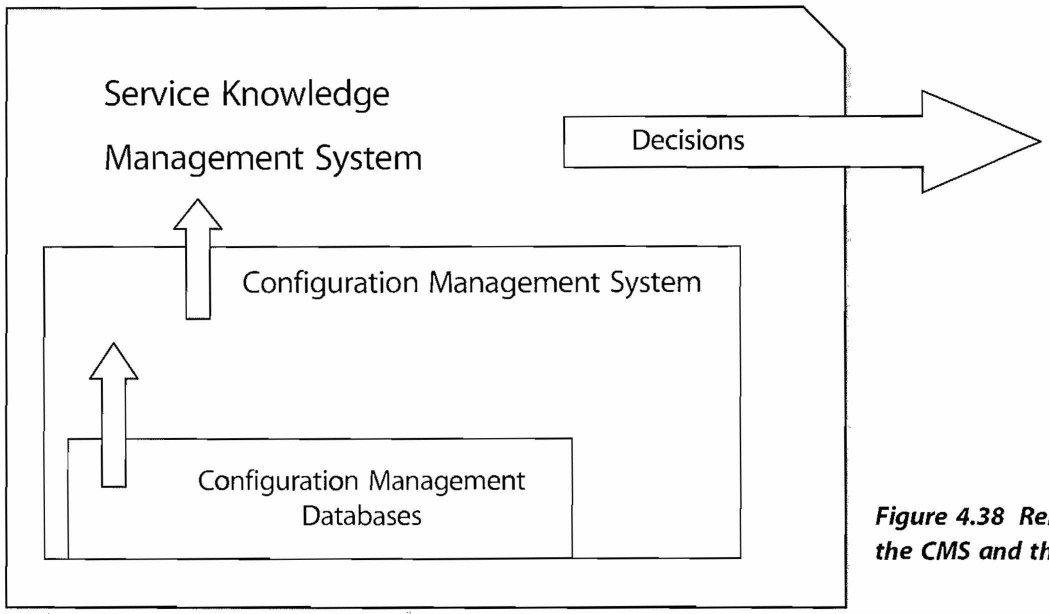


Figure 4.38 Relationship of the CMDB, the CMS and the SKMS

4.7.5 Process activities, methods and techniques

4.7.5.1 Knowledge Management strategy

An overall strategy for Knowledge Management is required. Where there is an organizational approach to Knowledge Management, initiatives within Service Transition, IT Service Management or other groupings should be designed to fit within the overall organizational approach.

In the absence of an organizational Knowledge Management approach, appropriate steps to establish Knowledge Management within Service Transition or within IT Service Management will be required. But even in this case developments should always be established with a view to as wide as practicable a span of Knowledge Management – covering direct IT staff, users, third party support and others likely to contribute or make beneficial use of the knowledge.

The strategy – either in place in the wider organization or being developed – will address:

- The governance model
- Organizational changes underway and planned and consequential changes in roles and responsibilities
- Establishing roles and responsibilities and ongoing funding
- Policies, processes, procedures and methods for Knowledge Management
- Technology and other resource requirements
- Performance measures.

Knowledge identification capture and maintenance

Specifically the strategy will identify and plan for the capture of relevant knowledge and the consequential information and data that will support it. The steps to delivering this include:

- Assisting an organization to identify knowledge that will be useful
- Designing a systematic process for organizing, distilling, storing and presenting information in a way that improves people's comprehension in a relevant area
- Accumulating knowledge through processes and workflow
- Generating new knowledge
- Accessing valuable knowledge from outside sources
- Capturing external knowledge and adapting it – data, information and knowledge from diverse sources such as databases, websites, employees, suppliers and partners.

4.7.5.2 Knowledge transfer

During the service lifecycle an organization needs to focus on retrieving, sharing and utilizing their knowledge through problem solving, dynamic learning, strategic planning and decision making. To achieve this, knowledge needs to be transferred to other parts of the organization at specific points in the lifecycle. Many of the Service Management processes will link into this, for example allowing the service desk to have optimum knowledge and understanding at the point for any Service Transition into support. They will be reliant on information sourced from release management such as known errors going into production but which are not show stoppers for the release schedule, or known error scripts from any of the technical support teams. Links with HR, facilities and other supporting services need to be established, maintained and utilized.

The challenge is often the practical problem of getting a knowledge package from one part of the organization to other parts of the organization. It is more than just sending an e-mail! Knowledge transfer is more complex; more accurately it is the activity through which one unit (e.g. a group, department or division) is affected by the experience of another. Its form must be applicable for those using it, and achieve a positive rating of 'ease of use'. The transfer of knowledge can be observed through changes in the knowledge or performance of recipients, at an individual or unit level.

An analysis of the knowledge gap (if any) within the organization should be undertaken. The gap will need to be researched and established by direct investigation of staff's understanding of the knowledge requirements for them to deliver their responsibilities compared with their actual observed knowledge. This can be a difficult task to deliver objectively and, rather than risk resentment or suspicion, it is often worth seeking skilled and experienced support to build this. The output from the knowledge gap exercise will form the basis for a communications improvement plan which will enable planning and measurement of success in communication of knowledge.

Traditionally knowledge transfer has been based on formal classroom training and documentation. In many cases the initial training is provided to a representative from a work group who is then required to cascade the knowledge to their working colleagues. Other techniques are often appropriate and form useful tools in the Service Transition armoury. Techniques worth considering include the following.

Learning styles

Different people learn in different ways, and the best method of transferring and maintaining knowledge within the Service Management and user community will need to be established. Learning styles vary with age, culture, attitude and personality. IT staff can be usefully reminded, especially where they are supporting users in a different working style, e.g. graphics design, performers, sales teams, that merely because a knowledge transfer mechanism works for them, it may not be appropriate for their current user base.

For many some element of 'hands-on' experience is a positive support for learning, and simulation exercises can be a useful consideration, or supervised experience and experimentation.

Knowledge visualization

This aims to improve the transfer of knowledge by using computer and non-computer-based visuals such as diagrams, images, photographs and storyboards. It focuses on the transfer of knowledge between people and aims to transfer insights, experiences, attitudes, values, expectations, perspectives, opinions and predictions by using various complementary visualizations. Dynamic forms of visualization such as educational animation have the potential to enhance understandings of systems that change over time. For example this can be particularly useful during a hardware refresh when the location of a component may change on an item, although the functionality does not alter.

Driving behaviour

Knowledge transfer aims to ensure that staff are able to decide on the correct actions to deliver their tasks in any foreseeable circumstances. For predictable and consistent tasks, the procedure can be incorporated within software tools that the staff use within those tasks. These procedures then drive behaviour in the accepted way. Change process models (see Figure 4.2) and service desk scripts are excellent examples. This includes the ability to recognize when the laid down practices are or might be inappropriate, e.g. in unexpected circumstances, when staff will either move away from the laid down rules when they do not deliver as required or else will escalate the situation.

Seminars, Webinars and advertising

Formally launching a new or changed service can create an 'event' that enhances the transfer of knowledge. Technology-based events such as Webinars offer the ability to deliver a high profile knowledge delivery mechanism with the ability to retain it online and deliver it subsequently to other locations and new staff. Internet and intranet portals can deliver equivalent messages in an ongoing fashion and allow discussion forums to question and develop knowledge.

Journals and newsletters

Regular communicating channels, once established, are useful in allowing knowledge to be transferred in smaller units – incrementally rather than 'big bang' can be easier to absorb and retain. They also allow for progressive training and adaptation to circumstance and time periods. Crucially these techniques can be made entertaining and targeted at specific groupings.

Aimed at the audience

A stock control system was introduced with staff in the warehouses directly inputting and working with the new system. Initially all documentation was formal and written in semi-technical terms and the staff taught how to use the system via traditional training and coaching. Once the system had settled in a monthly newsletter was planned to keep staff aware of changes, improvements, hints, tips etc. The first versions were, again, formal and addressed the required information only. It quickly became clear that the required knowledge was not in place within the staff. Success followed when the updates evolved into a genuine newsletter – among competitions, holiday snaps, humorous and even satirical articles the required user knowledge was transferred much more successfully. The lesson was that by targeting communications accurately at a known and understood audience, and making the experience pleasant, the required knowledge transfers along with the rest. And as a bonus the staff contributed entertaining articles and hints and tips they had evolved.

4.7.5.3 Data and information management

Knowledge rests on the management of the information and data that underpins it. To be efficient this process requires an understanding of some key process inputs such as how the data and information will be used:

- What knowledge is necessary based on what decisions must be made
- What conditions need to be monitored (changing external and internal circumstances, ranging from end-user demand, legal requirements through to weather forecasts)
- What data is available (what could be captured), as well as rejecting possible data capture as infeasible; this input may trigger justification for expenditure or changes in working practices designed to facilitate the capture of relevant data that would otherwise not be available

- The cost of capturing and maintaining data, and the value that data is likely to bring, bearing in mind the negative impact of data overload on effective knowledge transfer
- Applicable policies, legislation, standards and other requirements
- Intellectual property rights and copyright issues.

Successful data and information management will deliver:

- Conformance with legal and other requirements, e.g. company policy, codes of professional conduct
- Defined forms of data and information in a fashion that is easily usable by the organization
- Data and information that is current, complete and valid
- Data and information disposed of as required
- Data and information to the people who need it when they need it.

Establishing data and information requirements

The following activities should be planned and implemented in accordance with applicable organization policies and procedures with respect to the data and information management process. This plan and design is the responsibility of Service Strategy and Service Design.

Often, data and information is collected with no clear understanding of how it will be used and this can be costly. Efficiency and effectiveness are delivered by establishing the requirements for information. Sensible considerations, within the constraints determined as described above, might include:

- Establishing the designated data and information items, their content and form, together with the reason, e.g. technical, project, organizational, Service Management process, agreement, operations and information; data is costly to collect and often even more expensive to maintain, and so should be collected only when needed
- Encouraging the use of common and uniform content and format requirements to facilitate better and faster understanding of the content and help with consistent management of the data and information resources
- Establishing the requirements for data protection, privacy, security, ownership, agreement restrictions, rights of access, intellectual property and patents with the relevant stakeholder
- Defining who needs access to what data and information as well as when they access it, including the relative importance of it at different times. For example access to payroll information might be considered more important in the day before payroll is run than at other times of the month
- Considering any changes to the Knowledge Management process through Change Management.

Define the information architecture

In order to make effective use of data, in terms of delivering the required knowledge, a relevant architecture matched to the organizational situation and the knowledge requirements is essential. This in turn rests on:

- Creating and regularly updating a Service Management information model that enables the creation, use and sharing of information that is flexible, timely and cost-effective
- Defining systems that optimize the use of the information while maintaining data and information integrity
- Adopting data classification schemes that are in use across the organization, and if necessary negotiating changes to enable them to deliver within the Service Management area; where such organization-wide (or supply chain or industry sector) schemes do not exist, data classification schemes derived for use within Service Management should be designed with the intention of their being applicable across the organization to facilitate support for future organization-wide Knowledge Management.

An example of a knowledge, information and data architecture is shown in Figure 4.39.

Establishing data and information management procedures

When the requirements and architecture have been set up, data and information management to support Knowledge Management can be established. The key steps required involve setting up mechanisms to:

- Identify the service lifecycle data and information to be collected
- Define the procedure required to maintain the data and information and make it available to those requiring it
- Store and retrieve
- Establish authority and responsibility for all required items of information
- Define and publicize rights, obligations and commitments regarding the retention of, transmission of and access to information and data items based on applicable requirements and protecting its security, integrity and consistency

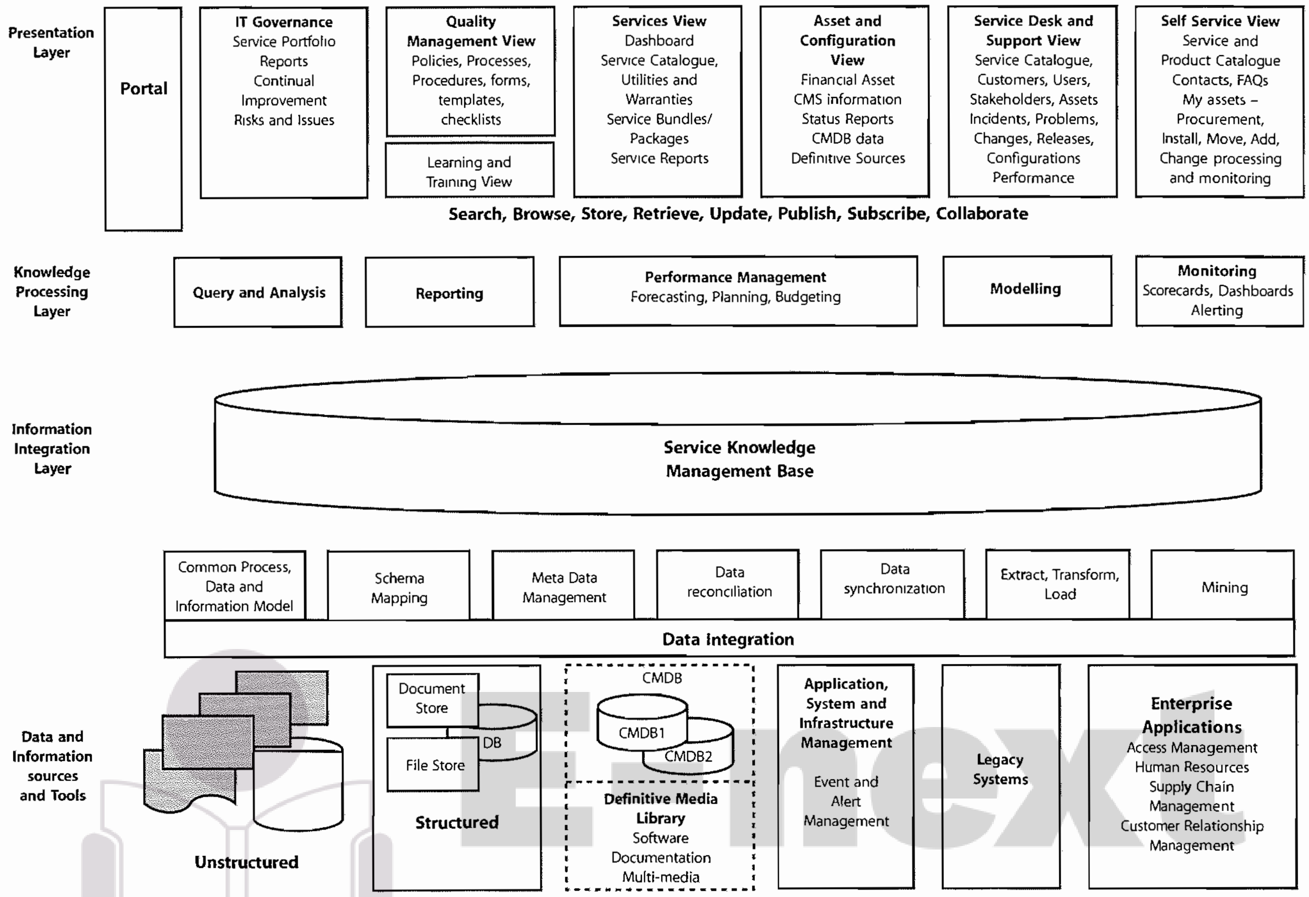

Presentation Layer
Portal
IT Governance
Service Portfolio
Reports
Continual Improvement
Risks and Issues
Quality Management View
Policies, Processes, Procedures, forms, templates, checklists
Learning and Training View
Services View
Dashboard
Service Catalogue, Utilities and Warranties
Service Bundles/ Packages
Service Reports
Asset and Configuration View
Financial Asset
CMS information
Status Reports
CMDB data
Definitive Sources
Service Desk and Support View
Service Catalogue, Customers, Users, Stakeholders, Assets
Incidents, Problems, Changes, Releases, Configurations
Performance
Self Service View
Service and Product Catalogue
Contacts, FAQs
My assets – Procurement, Install, Move, Add, Change processing and monitoring
Search, Browse, Store, Retrieve, Update, Publish, Subscribe, Collaborate
Knowledge Processing Layer
Query and Analysis
Reporting
Performance Management
Forecasting, Planning, Budgeting
Modelling
Monitoring
Scorecards, Dashboards
Alerting
Information Integration Layer
Service Knowledge Management Base
Common Process, Data and Information Model
Schema Mapping
Meta Data Management
Data reconciliation
Data synchronization
Extract, Transform, Load
Mining
Data Integration
Data and Information sources and Tools
Unstructured
Document Store
DB
File Store
Structured
CMDB
CMDB1
CMDB2
Definitive Media Library
Software
Documentation
Multi-media
Application, System and Infrastructure Management
Event and Alert Management
Legacy Systems
Enterprise Applications
Access Management
Human Resources
Supply Chain Management
Customer Relationship Management

Figure 4.39 Service knowledge management system

- Establish adequate backup and recovery of data and information; this should address reinstating the ability to make constructive use of information, not just the re-establishment of a database
- Identify the requirements to review, in the light of changing technology, organizational requirements, evolving policy and legislation (and if necessary to adapt to) changes in:
    - information system infrastructure in the light of evolving hardware and software technology
    - security, service continuity, storage and capacity
- Deal with collection and retention requirements.

When the procedures are designed, promulgated and accepted the organization can:

- Implement mechanisms to capture, store and retrieve the identified data from the relevant sources
- Manage the data and information storage and movement, especially in line with appropriate legislation.
- Archive designated information, in accordance with the data and information management plan including safely disposing of unwanted, invalid or unverifiable information according to the organization policy.

Evaluation and improvement

As with all processes, the capture and usage of data and information to support Knowledge Management and decision making requires attention to ongoing improvement, and the service improvement plan will take as relevant input:

- Measurement of the use made of the data and information management–data transactions
- Evaluation of the usefulness of the data and information – identified by relevance of reports produced
- Identification of any data or information or registered users that no longer seem relevant to the organization's knowledge requirements.

4.7.5.4 Using the service knowledge management system

Providing services to customers across time zones, work cycles, and geographies requires good knowledge sharing across all locations and time periods of Service Operations. A service provider must first establish a service knowledge management system that can be shared, updated and used by its operating entities, partners, and customers. Figure 4.39 shows an example of the architecture for such a system.

Implementation of a service knowledge management system helps reduce the costs of maintaining and managing the services, both by increasing the efficiency of operational management procedures and by reducing the risks that arise from the lack of proper mechanisms.

All training and knowledge material needs to be aligned to the business perspective. Materials that can be included are:

- The business language and terminology and how IT terminology is translated
- The business processes and where IT underpins them
- Any SLAs, and supporting agreements and contracts that would change as a result of the new Service Transition – this is especially important for the service desk analysts whose target at support transition will be to sustain service; if classifications are accurate this will facilitate the whole process.

For those in the Service Transition process a good way of consolidating understanding is to either spend time in the development areas, taking part in some of the testing processes, or to spend time in the business at the receiving end of the Service Transition to understand the process from the business perspective.

Case study

Current situation An organization analysed that at least 75% of the cost of delivering support comes from resolving customer issues. It was using point technologies such as a service desk workflow tool, search engines, scripting tools or simple knowledge bases. These systems generally focused parts of the resolution process and they were not very effective. This contributed to dissatisfied customers, resulted in an ineffective service desk and caused integration issues for IT.

Solution A comprehensive SKMS was implemented to help to address these obstacles by combining intelligent search and Knowledge Management with Service Management and business process support, authoring workflows and comprehensive self-service facilities.

The SKMS was supported by the problem management and Change Management process.

The experience of end users who come to the website for help was dramatically improved. Instead of an empty search box followed by no results or far too many, the application leads the user through a structured set of steps. Based on the specifics of the incident or request and the customer, web screens will guide users to specific answers, follow-up questions, escalation options, opportunities to drill down or just highly relevant search results. The following improvements were achieved:

- Increased agent productivity
- Reduced aversion to web self-service
- Fewer escalations.

Over time the web workflows were tuned to deliver more and more optimized experiences. Good experiences helped to add value to the product and services and this resulted in greater loyalty that in turn increased profits.

Conclusion A wealth of information exists in most organizations that is not initially thought to contribute to the decision process, but, when used as supplemental to traditional configuration data, can bring the lessons of history into sharp focus. Often this information is in an informal fashion. Marketing, sales, customer and staff information is a commonly overlooked source of valuable trend data that, along with traditional configuration, can paint a larger, more meaningful picture of the landscape and uncover the right 'course corrections' to bring a Service Transition or operational support for a service back on track and keep an organization travelling towards its objectives. Without this clear picture, the effectiveness diminishes and efficiency will decay. By recognizing that this is in place, organizations can more easily justify the resource costs of establishing and maintaining the data, processes, knowledge and skills needed to make it as effective as possible and maximize the benefits.

Useful materials include:

- Process maps to understand all the integrated activities
- Any known error logs and the workarounds – again particularly important for the service desk
- Business and other public calendars.

Technology for service desks and customer service needs to make it easier for customers, users and service desk agents. Some minimal progress has been made with generic Knowledge Management tools and there are significant developments in the Service Management industry to develop mature, process-oriented business applications supported by comprehensive knowledge bases. Examples of potential benefits are:

- **Agent efficiency** – The largest component of ROI from Knowledge Management is reduced incident handling time and increased agent productivity.
- **Self-service** – A comprehensive SKMS provides the customer with knowledge directly on the support website. The cost of self-service is an order of magnitude lower than assisted service.

4.7.6 Triggers, inputs and outputs and inter-process interfaces

Crucial to Knowledge Management is the need to ensure that the benefits of Knowledge Management are understood and enthusiastically embraced within the whole organization. Specifically, effective Knowledge Management depends on the committed support and delivery by most, if not all, of those working in and around IT Service Management.

Service Operations

Errors within the service detected during transition will be recorded and analysed and the knowledge about their existence, consequences and workarounds will be made available to Service Operations in an easy to use fashion.

Operations staff

- Front-line incident management staff, on service desk and second-line support, are the point of capture for much of the everyday IT Service Management data. If these staff do not understand the importance of their role then Knowledge Management will not be effective. Traditionally support analysts have been reluctant to record their actions fully, feeling that this can undermine their position within the organization – allowing issues to be resolved without them. Changing this to an attitude of appreciating the benefits – to individuals and the organization – of widely re-usable knowledge is the key to successful Knowledge Management.
- Problem management staff will be key users of collected knowledge and typically responsible for the normalization of data capture by means of developing and maintaining scripts supporting data capture within incident management.

Transition staff

Service Transition staff capture data of relevance through all lifecycle phases and so need to be aware of the importance of collecting it accurately and completely. Service Transition staff capture data and information:

- Relevant to adaptability and accessibility of the service as designed, to be fed back, via CSI, to Service Design
- 'Course corrections' and other adaptations to the design required during transition. Awareness and understanding of these will make subsequent transitions easier.

4.7.7 Key performance indicators and metrics

A strong Business Case is critical for effective Knowledge Management and it is important that the measures of success are visible to all levels involved in the implementation.

Typical measures for an IT service provider's contribution are:

- Successful implementation and early life operation of new and changed services with few knowledge-related errors
- Increased responsiveness to changing business demands, e.g. higher percentage of queries and question solved via single access to internet/intranet through use of search and index systems such as Google
- Improved accessibility and management of standards and policies
- Knowledge dissemination
- Reduced time and effort required to support and maintain services
- Reduced time to find information for diagnosis and fixing incidents and problems
- Reduced dependency on personnel for knowledge.

4.7.7.1 Evaluation and improvement

Although hard to measure the value of knowledge, it is nonetheless important to determine the value to the organization in order to ensure the case for expenditure and support of Knowledge Management is maintainable. The costs associated with Knowledge Management can then be measured and compared against that value.

4.7.7.2 Indicators relevant to business/customers

Knowledge Management is an enabling process and so demonstration of its effectiveness needs to be inferred from indirect measurement. Elements of the service quality that will be positively influenced by good Knowledge Management might include:

- Reduction in the 'user error' category of errors due to targeted knowledge transfer, coupled with cheaper user training costs
- Lower incident, problem and error resolution times influenced by better targeted support staff training and by a relevant, maintained and accessible knowledge base containing workarounds
- Enhanced customer experiences such as:
  - Quicker resolution of a query
  - The ability to solve issues directly without external support
  - Less transfer of issues to other people and resolution at lower staff levels
- Reduced time for transition and duration of early life support.

Measuring benefit from knowledge transfer

The value of improved knowledge transfer during Service Transition through improved Knowledge Management can be measured via the increased effectiveness of staff using and supporting the new or changed service. This (effectively the steepness of the learning curve) in turn can be measured through:

- Incidents and lost time categorized as 'lack of user knowledge'
- Average diagnosis and repair time for faults fixed in-house
- Incidents related to new or changed services fixed by reference to knowledge base.

Although not every element of the above can be directly attributable to Knowledge Management, the trends in these measures will be influenced by the quality of Knowledge Management, as shown by the example in Figure 4.40.

Clearly, the performance of the support groups post transition will be a determining factor of the quality of the knowledge transfer, typically delivered via training; however, it is more proactive to check understanding before arriving at this point. After each piece of training activity there should be a feedback mechanism to check understanding and quality of delivery. This could be in the form of a post course questionnaire, or even a test to confirm understanding.

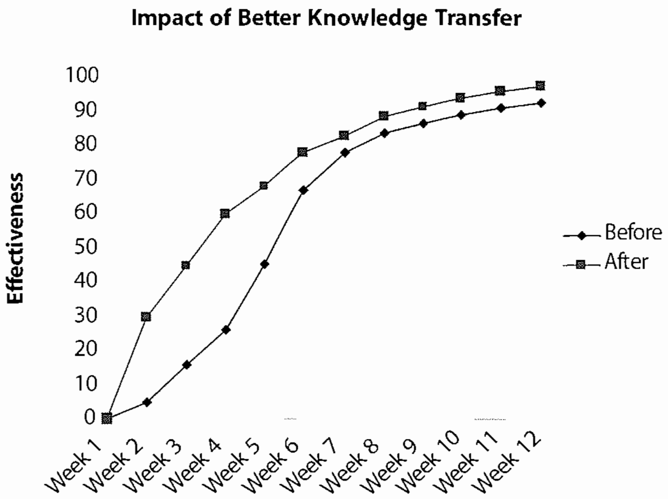


Figure 4.40 Contribution of knowledge to effectiveness of support staff

4.7.7.3 Measures directly relevant to the service provider

Indications of the effectiveness of the Knowledge Management process itself include:

- Usage of the knowledge base, measured by:
  - Number of accesses to the SKMS
  - Average time taken to find materials
- Errors reported by staff or detected at audit (none probably means no one was using it)
- Involvement of staff in discussion/query/answer forums providing support through knowledge sharing and capture of that shared knowledge
- Degree of re-use of material in documentation such as procedures, test design and service desk scripts
- Satisfaction with training courses, newsletters, web briefings etc.