4 Service Operation processes

The processes listed in paragraph 2.4.5 are discussed in detail in this chapter. As a reference, the overall structure is briefly described here and then each of the processes is described in more detail later in the chapter. Please note that the roles for each process and the tools used for each process are described in Chapters 6 and 7 respectively.

- Event Management is the process that monitors all events that occur through the IT Infrastructure to allow for normal operation and also to detect and escalate exception conditions.
- Incident Management concentrates on restoring the service to users as quickly as possible, in order to minimize business impact.
- Problem Management involves root-cause analysis to determine and resolve the cause of events and incidents, proactive activities to detect and prevent future problems/incidents and a Known Error sub-process to allow quicker diagnosis and resolution if further incidents do occur.

NOTE: Without this distinction between incidents and problems, and keeping separate Incident and Problem Records, there is a danger that either:

  - Incidents will be closed too early in the overall support cycle and there will be no actions taken to prevent recurrence – so the same incidents will have to be fixed over and over again, or
  - Incidents will be kept open so that root cause analysis can be done and visibility will be lost of when the user's service was actually restored – so SLA targets may not be met even though the service has been restored within users' expectations. This often results in a large number of open incidents, many of which will never be closed unless a periodic 'purge' is undertaken. This can be very de-motivating and can prevent effective visibility of current issues.

- Request Fulfilment involves the management of customer or user requests that are not generated as an incident from an unexpected service delay or disruption. Some organizations may choose to handle such requests as a 'category' of incidents and manage the information through an Incident Management system – but others may choose (because of high volumes or business priority of such requests) to facilitate the provision of Request Fulfilment capabilities separately via the Request Fulfilment process. It has become popular practice to use a formal Request Fulfilment process to manage customer and user requests for all types of requests which include facilities, moves and supplies as well as those specific to IT services. These requests are not generally tied to the same SLA measures and separating the records and the process flow is emerging as best practice in many organizations.

- Access Management: this is the process of granting authorized users the right to use a service, while restricting access to non-authorized users. It is based on being able accurately to identify authorized users and then manage their ability to access services as required during different stages of their human resources (HR) or contractual lifecycle. Access Management has also been called Identity or Rights Management in some organizations.

In addition, there are several other processes that will be executed or supported during Service Operation, but which are driven during other phases of the Service Management Lifecycle. The operational aspects of these processes will be discussed in the final part of this chapter and include:

- Change Management, a major process which should be closely linked to Configuration Management and Release Management. These topics are primarily covered in the Service Transition Publication.
- Capacity and Availability Management, the operational aspects of which are covered in this publication, but which are covered in more detail in the Service Design publication.
- Financial Management, which is covered in the Service Strategy publication.
- Knowledge Management, which is covered in the Service Transition publication.
- IT Service Continuity, which is covered in the Service Design publication.
- Service Reporting and Measurement, which are covered in the Continual Service Improvement publication

4.1 Event Management

An event can be defined as any detectable or discernible occurrence that has significance for the management of the IT Infrastructure or the delivery of IT service and evaluation of the impact a deviation might cause to the services. Events are typically notifications created by an IT service, Configuration Item (CI) or monitoring tool.

Effective Service Operation is dependent on knowing the status of the infrastructure and detecting any deviation from normal or expected operation. This is provided by good monitoring and control systems, which are based on two types of tools:

- active monitoring tools that poll key CIs to determine their status and availability. Any exceptions will generate an alert that needs to be communicated to the appropriate tool or team for action
- passive monitoring tools that detect and correlate operational alerts or communications generated by CIs.

4.1.1 Purpose/goal/objective

The ability to detect events, make sense of them and determine the appropriate control action is provided by Event Management. Event Management is therefore the basis for Operational Monitoring and Control (see Appendix B).

In addition, if these events are programmed to communicate operational information as well as warnings and exceptions, they can be used as a basis for automating many routine Operations Management activities, for example executing scripts on remote devices, or submitting jobs for processing, or even dynamically balancing the demand for a service across multiple devices to enhance performance.

Event Management therefore provides the entry point for the execution of many Service Operation processes and activities. In addition, it provides a way of comparing actual performance and behaviour against design standards and SLAs. As such, Event Management also provides a basis for Service Assurance and Reporting; and Service Improvement. This is covered in detail in the Continual Service Improvement publication.

4.1.2 Scope

Event Management can be applied to any aspect of Service Management that needs to be controlled and which can be automated. These include:

- Configuration Items:

    - Some CIs will be included because they need to stay in a constant state (e.g. a switch on a network needs to stay on and Event Management tools confirm this by monitoring responses to 'pings').
    - Some CIs will be included because their status needs to change frequently and Event Management can be used to automate this and update the CMS (e.g. the updating of a file server).
- Environmental conditions (e.g. fire and smoke detection)
- Software licence monitoring for usage to ensure optimum/legal licence utilization and allocation
- Security (e.g. intrusion detection)
- Normal activity (e.g. tracking the use of an application or the performance of a server).

The difference between monitoring **and** Event Management

These two areas are very closely related, but slightly different in nature. Event Management is focused on generating and detecting meaningful notifications about the status of the IT Infrastructure and services.

While it is true that monitoring is required to detect and track these notifications, monitoring is broader than Event Management. For example, monitoring tools will check the status of a device to ensure that it is operating within acceptable limits, even if that device is not generating events.

Put more simply, Event Management works with occurrences that are specifically generated to be monitored. Monitoring tracks these occurrences, but it will also actively seek out conditions that do not generate events.

4.1.3 Value to business

Event Management's value to the business is generally indirect; however, it is possible to determine the basis for its value as follows:

- Event Management provides mechanisms for early detection of incidents. In many cases it is possible for the incident to be detected and assigned to the appropriate group for action before any actual service outage occurs.
- Event Management makes it possible for some types of automated activity to be monitored by exception – thus removing the need for expensive and resource intensive real-time monitoring, while reducing downtime.
- When integrated into other Service Management processes (such as, for example, Availability or Capacity Management), Event Management can signal status changes or exceptions that allow the appropriate person or team to perform early response, thus improving the performance of the process. This, in turn, will allow the business to benefit from more effective and more efficient Service Management overall.

- Event Management provides a basis for automated operations, thus increasing efficiencies and allowing expensive human resources to be used for more innovative work, such as designing new or improved functionality or defining new ways in which the business can exploit technology for increased competitive advantage.

4.1.4 Policies/principles/basic concepts

There are many different types of events, for example:

- Events that signify regular operation:
  - notification that a scheduled workload has completed
  - a user has logged in to use an application
  - an e-mail has reached its intended recipient.
- Events that signify an exception
  - a user attempts to log on to an application with the incorrect password
  - an unusual situation has occurred in a business process that may indicate an exception requiring further business investigation (e.g. a web page alert indicates that a payment authorization site is unavailable – impacting financial approval of business transactions)
  - a device's CPU is above the acceptable utilization rate
  - a PC scan reveals the installation of unauthorized software.
- Events that signify unusual, but not exceptional, operation. These are an indication that the situation may require closer monitoring. In some cases the condition will resolve itself, for example in the case of an unusual combination of workloads – as they are completed, normal operation is restored. In other cases, operator intervention may be required if the situation is repeated or if it continues for too long. These rules or policies are defined in the Monitoring and Control Objectives for that device or service. Examples of this type of event are:
  - A server's memory utilization reaches within 5% of its highest acceptable performance level
  - The completion time of a transaction is 10% longer than normal.

Two things are significant about the above examples:

- Exactly what constitutes normal versus unusual operation, versus an exception? There is no definitive rule about this. For example, a manufacturer may provide that a benchmark of 75% memory utilization is optimal for application X. However, it is discovered that, under the specific conditions of our organization, response times begin to degrade above 70% utilization. The next section will explore how these figures are determined.
- Each relies on the sending and receipt of a message of some type. These are generally referred to as Event notifications and they don't just happen.

The next paragraphs will explore exactly how events are defined, generated and captured.

4.1.5 Process activities, methods and techniques

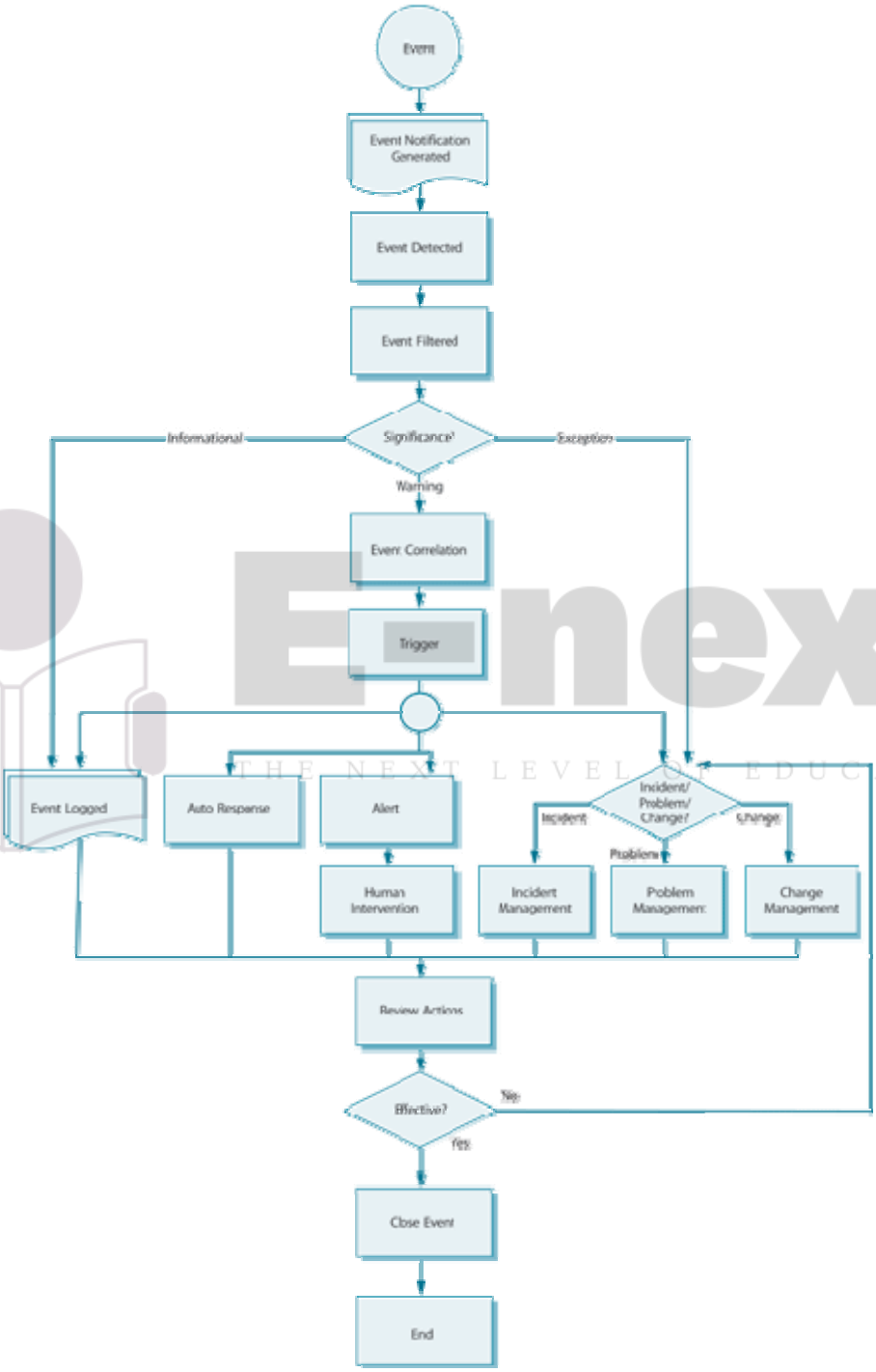

Figure 4.1 The Event Management process

Figure 4.1 is a high-level and generic representation of Event Management. It should be used as a reference and definition point, rather than an actual Event Management flowchart. Each activity in this process is described below.

4.1.5.1 Event occurs

Events occur continuously, but not all of them are detected or registered. It is therefore important that everybody involved in designing, developing, managing and supporting IT services and the IT Infrastructure that they run on understands what types of event need to be detected.

This is discussed in paragraph 4.1.10.1, titled ‘Instrumentation’.

4.1.5.2 Event notification

Most CIs are designed to communicate certain information about themselves in one of two ways:

- A device is interrogated by a management tool, which collects certain targeted data. This is often referred to as polling.
- The CI generates a notification when certain conditions are met. The ability to produce these notifications has to be designed and built into the CI, for example a programming hook inserted into an application.

Event notifications can be proprietary, in which case only the manufacturer’s management tools can be used to detect events. Most CIs, however generate Event notifications using an open standard such as SNMP (Simple Network Management Protocol).

Many CIs are configured to generate a standard set of events, based on the designer’s experience of what is required to operate the CI, with the ability to generate additional types of event by ‘turning on’ the relevant event generation mechanism. For other CI types, some form of ‘agent’ software will have to be installed in order to initiate the monitoring. Often this monitoring feature is free, but sometimes there is a cost to the licensing of the tool.

In an ideal world, the Service Design process should define which events need to be generated and then specify how this can be done for each type of CI. During Service Transition, the event generation options would be set and tested.

In many organizations, however, defining which events to generate is done by trial and error. System managers use the standard set of events as a starting point and then tune the CI over time, to include or exclude events as required. The problem with this approach is that it only takes into account the immediate needs of the staff managing the device and does not facilitate good planning or improvement. In addition, it makes it very difficult to monitor and manage the

service over all devices and staff. One approach to combating this problem is to review the set of events as part of continual improvement activities.

A general principle of Event notification is that the more meaningful the data it contains and the more targeted the audience, the easier it is to make decisions about the event. Operators are often confronted by coded error messages and have no idea how to respond to them or what to do with them. Meaningful notification data and clearly defined roles and responsibilities need to be articulated and documented during Service Design and Service Transition (see also paragraph 4.1.10.1 on 'Instrumentation'). If roles and responsibilities are not clearly defined, in a wide alert, no one knows who is doing what and this can lead to things being missed or duplicated efforts.

4.1.5.3 Event detection

Once an Event notification has been generated, it will be detected by an agent running on the same system, or transmitted directly to a management tool specifically designed to read and interpret the meaning of the event.

4.1.5.4 Event filtering

The purpose of filtering is to decide whether to communicate the event to a management tool or to ignore it. If ignored, the event will usually be recorded in a log file on the device, but no further action will be taken.

The reason for filtering is that it is not always possible to turn Event notification off, even though a decision has been made that it is not necessary to generate that type of event. It may also be decided that only the first in a series of repeated Event notifications will be transmitted.

During the filtering step, the first level of correlation is performed, i.e. the determination of whether the event is informational, a warning, or an exception (see next step). This correlation is usually done by an agent that resides on the CI or on a server to which the CI is connected.

The filtering step is not always necessary. For some CIs, every event is significant and moves directly into a management tool's correlation engine, even if it is duplicated. Also, it may have been possible to turn off all unwanted Event notifications.

4.1.5.5 Significance of events

Every organization will have its own categorization of the significance of an event, but it is suggested that at least these three broad categories be represented:

- **Informational:** This refers to an event that does not require any action and does not represent an exception. They are typically stored in the system or service log files and kept for a predetermined period. Informational events are typically used to check on the status of a device or service, or to confirm the successful completion of an activity. Informational events can also be used to generate statistics (such as the number of users logged on to an application during a certain period) and as input into investigations (such as which jobs completed successfully before the transaction processing queue hung). Examples of informational events include:
  - A user logs onto an application
  - A job in the batch queue completes successfully
  - A device has come online
  - A transaction is completed successfully.
- **Warning:** A warning is an event that is generated when a service or device is approaching a threshold. Warnings are intended to notify the appropriate person, process or tool so that the situation can be checked and the appropriate action taken to prevent an exception. Warnings are not typically raised for a device failure. Although there is some debate about whether the failure of a redundant device is a warning or an exception (since the service is still available). A good rule is that every failure should be treated as an exception, since the risk of an incident impacting the business is much greater. Examples of warnings are:
  - Memory utilization on a server is currently at 65% and increasing. If it reaches 75%, response times will be unacceptably long and the OLA for that department will be breached.
  - The collision rate on a network has increased by 15% over the past hour.
- **Exception:** An exception means that a service or device is currently operating abnormally (however that has been defined). Typically, this means that an OLA and SLA have been breached and the business is being impacted. Exceptions could represent a total failure, impaired functionality or degraded performance. Please note, though, that an exception does not always represent an incident. For example, an exception could be generated when an unauthorized device is discovered on the network. This can be managed by using either an Incident Record or a Request for Change (or even both), depending on the organization's Incident and Change Management policies. Examples of exceptions include:
  - A server is down
  - Response time of a standard transaction across the network has slowed to more than 15 seconds
  - More than 150 users have logged on to the General Ledger application concurrently
  - A segment of the network is not responding to routine requests.

4.1.5.6 Event correlation

If an event is significant, a decision has to be made about exactly what the significance is and what actions need to be taken to deal with it. It is here that the meaning of the event is determined.

Correlation is normally done by a 'Correlation Engine', usually part of a management tool that compares the event with a set of criteria and rules in a prescribed order. These criteria are often called Business Rules, although they are generally fairly technical. The idea is that the event may represent some impact on the business and the rules can be used to determine the level and type of business impact.

A Correlation Engine is programmed according to the performance standards created during Service Design and any additional guidance specific to the operating environment.

Examples of what Correlation Engines will take into account include:

- Number of similar events (e.g. this is the third time that the same user has logged in with the incorrect password, a business application reports that there has been an unusual pattern of usage of a mobile telephone that could indicate that the device has been lost or stolen)
- Number of CIs generating similar events
- Whether a specific action is associated with the code or data in the event
- Whether the event represents an exception
- A comparison of utilization information in the event with a maximum or minimum standard (e.g. has the device exceeded a threshold?)
- Whether additional data is required to investigate the event further, and possibly even a collection of that data by polling another system or database
- Categorization of the event
- Assigning a priority level to the event.

4.1.5.7 Trigger

If the correlation activity recognizes an event, a response will be required. The mechanism used to initiate that response is called a trigger.

There are many different types of triggers, each designed specifically for the task it has to initiate. Some examples include:

- Incident Triggers that generate a record in the Incident Management system, thus initiating the Incident Management process
- Change Triggers that generate a Request for Change (RFC), thus initiating the Change Management process

- A trigger resulting from a approved RFC that has been implemented but caused the event, or from an unauthorized change that has been detected – in either case this will be referred to Change Management for investigation
- Scripts that execute specific actions, such as submitting batch jobs or rebooting a device
- Paging systems that will notify a person or team of the event by mobile phone
- Database triggers that restrict access of a user to specific records or fields, or that create or delete entries in the database.

4.1.5.8 Response selection

At this point in the process, there are a number of response options available. It is important to note that the response options can be chosen in any combination. For example, it may be necessary to preserve the log entry for future reference, but at the same time escalate the event to an Operations Management staff member for action.

The options in the flowchart are examples. Different organizations will have different options, and they are sure to be more detailed. For example, there will be a range of auto responses for each different technology. The process of determining which one is appropriate and how to execute it are not represented in this flowchart. Some of the options available are:

- **Event logged:** Regardless of what activity is performed, it is a good idea to have a record of the event and any subsequent actions. The event can be logged as an Event Record in the Event Management tool, or it can simply be left as an entry in the system log of the device or application that generated the event. If this is the case, though, there needs to be a standing order for the appropriate Operations Management staff to check the logs on a regular basis and clear instructions about how to use each log. It should also be remembered that the event information in the logs may not be meaningful until an incident occurs; and where the Technical Management staff use the logs to investigate where the incident originated. This means that the Event Management procedures for each system or team need to define standards about how long events are kept in the logs before being archived and deleted.
- **Auto response:** Some events are understood well enough that the appropriate response has already been defined and automated. This is normally as a result of good design or of previous experience (usually Problem Management). The trigger will initiate the action and then evaluate whether it was completed successfully. If not, an Incident or Problem Record will be created. Examples of auto responses include:
  - Rebooting a device
  - Restarting a service

- Submitting a job into batch
- Changing a parameter on a device
- Locking a device or application to protect it against unauthorized access.

Note: locking a device may result in denial of service to authorized users, which could be exploited by a deliberate attacker – so great care should be taken when deciding whether this is an appropriate automated action. Where this response is used it may be prudent to also combine this with a call for human intervention, so that the automated action can be swiftly checked and approved.

- Alert **and human intervention:** If the event requires human intervention, it will need to be escalated. The purpose of the alert is to ensure that the person with the skills appropriate to deal with the event is notified. The alert will contain all the information necessary for that person to determine the appropriate action – including reference to any documentation required (e.g. user manuals). It is important to note that this is not necessarily the same as the functional escalation of an incident, where the emphasis is on restoring service within an agreed time (which may require a variety of activities). The alert requires a person, or team, to perform a specific action, possibly on a specific device and possibly at a specific time, e.g. changing a toner cartridge in a printer when the level is low.
- Incident**,** problem **or** change**?** Some events will represent a situation where the appropriate response will need to be handled through the Incident, Problem or Change Management process. These are discussed below, but it is important to note that a single incident may initiate any one or a combination of these three processes – for example, a non-critical server failure is logged as an incident, but as there is no workaround, a Problem Record is created to determine the root cause and resolution and an RFC is logged to relocate the workload onto an alternative server while the problem is resolved.
- **Open an RFC:** There are two places in the Event Management process where an RFC can be created:
  - **When an exception occurs:** For example, a scan of a network segment reveals that two new devices have been added without the necessary authorization. A way of dealing with this situation is to open an RFC, which can be used as a vehicle for the Change Management process to deal with the exception (as an alternative to the more conventional approach of opening an incident that would be routed via the Service Desk to Change Management). Investigation by Change Management is appropriate here since unauthorized changes imply that the Change Management process was not effective.
  - **Correlation identifies that a** change **is needed:** In this case the event correlation activity determines that the appropriate response

to an event is for something to be changed. For example, a performance threshold has been reached and a parameter on a major server needs to be tuned. How does the correlation activity determine this? It was programmed to do so either in the Service Design process or because this has happened before and Problem Management or Operations Management updated the Correlation Engine to take this action.

- **Open an** Incident Record**:** As with an RFC, an incident can be generated immediately when an exception is detected, or when the Correlation Engine determines that a specific type or combination of events represents an incident. When an Incident Record is opened, as much information as possible should be included – with links to the events concerned and if possible a completed diagnostic script.
- **Open or link to a** Problem Record**:** It is rare for a Problem Record to be opened without related incidents (for example as a result of a Service Failure Analysis (see Service Design publication) or maturity assessment, or because of a high number of retry network errors, even though a failure has not yet occurred). In most cases this step refers to linking an incident to an existing Problem Record. This will assist the Problem Management teams to reassess the severity and impact of the problem, and may result in a changed priority to an outstanding problem.

  However, it is possible, with some of the more sophisticated tools, to evaluate the impact of the incidents and also to raise a Problem Record automatically, where this is warranted, to allow root-cause analysis to commence immediately.

- **Special types of incident:** In some cases an event will indicate an exception that does not directly impact any IT service, for example, a redundant air conditioning unit fails, or unauthorized entry to a data centre. Guidelines for these events are as follows:
  - An incident should be logged using an Incident Model that is appropriate for that type of exception, e.g. an Operations Incident or Security Incident (see paragraph 4.2.4.2 for more details of Incident Models).
  - The incident should be escalated to the group that manages that type of incident.
  - As there is no outage, the Incident Model used should reflect that this was an operational issue rather than a service issue. The statistics would not normally be reported to customers or users, unless they can be used to demonstrate that the money invested in redundancy was a good investment.
  - These incidents should not be used to calculate downtime, and can in fact be used to demonstrate how proactive IT has been in making services available.

4.1.5.9 Review actions

With thousands of events being generated every day, it is not possible formally to review every individual event. However, it is important to check that any significant events or exceptions have been handled appropriately, or to track trends or counts of event types, etc. In many cases this can be done automatically, for example polling a server that had been rebooted using an automated script to see that it is functioning correctly.

In the cases where events have initiated an incident, problem and/or change, the Action Review should not duplicate any reviews that have been done as part of those processes. Rather, the intention is to ensure that the handover between the Event Management process and other processes took place as designed and that the expected action did indeed take place. This will ensure that incidents, problems or changes originating within Operations Management do not get lost between the teams or departments.

The Review will also be used as input into continual improvement and the evaluation and audit of the Event Management process.

4.1.5.10 Close event

Some events will remain open until a certain action takes place, for example an event that is linked to an open incident. However, most events are not ‘opened’ or ‘closed’.

Informational events are simply logged and then used as input to other processes, such as Backup and Storage Management. Auto-response events will typically be closed by the generation of a second event. For example, a device generates an event and is rebooted through auto response – as soon as that device is successfully back online, it generates an event that effectively closes the loop and clears the first event.

It is sometimes very difficult to relate the open event and the close notifications as they are in different formats. It is optimal that devices in the infrastructure produce ‘open’ and ‘close’ events in the same format and specify the change of status. This allows the correlation step in the process to easily match open and close notifications.

In the case of events that generated an incident, problem or change, these should be formally closed with a link to the appropriate record from the other process.

4.1.6 Triggers, input and output/inter-process interfaces

Event Management can be initiated by any type of occurrence. The key is to define which of these occurrences is significant and which need to be acted upon. Triggers include:

- Exceptions to any level of CI performance defined in the design specifications, OLAs or SOPs
- Exceptions to an automated procedure or process, e.g. a routine change that has been assigned to a build team has not been completed in time
- An exception within a business process that is being monitored by Event Management
- The completion of an automated task or job
- A status change in a device or database record
- Access of an application or database by a user or automated procedure or job
- A situation where a device, database or application, etc. has reached a predefined threshold of performance.

Event Management can interface to any process that requires monitoring and control, especially those that do not require real-time monitoring, but which do require some form of intervention following an event or group of events. Examples of interfaces with other processes include:

- Interface with business applications and/or business processes to allow potentially significant business events to be detected and acted upon (e.g. a business application reports abnormal activity on a customer's account that may indicate some sort of fraud or security breach).
- The primary ITSM relationships are with Incident, Problem and Change Management. These interfaces are described in some detail in paragraph 4.1.5.8.
- Capacity and Availability Management are critical in defining what events are significant, what appropriate thresholds should be and how to respond to them. In return, Event Management will improve the performance and availability of services by responding to events when they occur and by reporting on actual events and patterns of events to determine (by comparison with SLA targets and KPIs) if there is some aspect of the infrastructure design or operation that can be improved.
- Configuration Management is able to use events to determine the current status of any CI in the infrastructure. Comparing events with the authorized baselines in the Configuration Management System (CMS) will help to determine whether there is unauthorized Change activity taking place in the organization (see Service Transition publication).
- Asset Management (covered in more detail in the Service Design and Transition publications) can use Event Management to determine the lifecycle status of assets. For example, an event could be generated to

signal that a new asset has been successfully configured and is now operational.

- Events can be a rich source of information that can be processed for inclusion in Knowledge Management systems. For example, patterns of performance can be correlated with business activity and used as input into future design and strategy decisions.
- Event Management can play an important role in ensuring that potential impact on SLAs is detected early and any failures are rectified as soon as possible so that impact on service targets is minimized.

4.1.7 Information Management

Key information involved in Event Management includes the following:

- SNMP messages, which are a standard way of communicating technical information about the status of components of an IT Infrastructure.
- Management Information Bases (MIBs) of IT devices. An MIB is the database on each device that contains information about that device, including its operating system, BIOS version, configuration of system parameters, etc. The ability to interrogate MIBs and compare them to a norm is critical to being able to generate events.
- Vendor's monitoring tools agent software.
- Correlation Engines contain detailed rules to determine the significance and appropriate response to events. Details on this are provided in paragraph 4.1.5.6.
- There is no standard Event Record for all types of event. The exact contents and format of the record depend on the tools being used, what is being monitored (e.g. a server and the Change Management tools will have very different data and probably use a different format). However, there is some key data that is usually required from each event to be useful in analysis. It should typically include the:
    - Device
    - Component
    - Type of failure
    - Date/time
    - Parameters in exception
    - Value.

4.1.8 Metrics

For each measurement period in question, the metrics to check on the effectiveness and efficiency of the Event Management process should include the following:

- Number of events by category
- Number of events by significance

- Number and percentage of events that required human intervention and whether this was performed
- Number and percentage of events that resulted in incidents or changes
- Number and percentage of events caused by existing problems or Known Errors. This may result in a change to the priority of work on that problem or Known Error
- Number and percentage of repeated or duplicated events. This will help in the tuning of the Correlation Engine to eliminate unnecessary event generation and can also be used to assist in the design of better event generation functionality in new services
- Number and percentage of events indicating performance issues (for example, growth in the number of times an application exceeded its transaction thresholds over the past six months)
- Number and percentage of events indicating potential availability issues (e.g. failovers to alternative devices, or excessive workload swapping)
- Number and percentage of each type of event per platform or application
- Number and ratio of events compared with the number of incidents.

4.1.9 Challenges, Critical Success Factors and risks

4.1.9.1 Challenges

There are a number of challenges that might be encountered:

- An initial challenge may be to obtain funding for the necessary tools and effort needed to install and exploit the benefits of the tools.
- One of the greatest challenges is setting the correct level of filtering. Setting the level of filtering incorrectly can result in either being flooded with relatively insignificant events, or not being able to detect relatively important events until it is too late.
- Rolling out of the necessary monitoring agents across the entire IT infrastructure may be a difficult and time-consuming activity requiring an ongoing commitment over quite a long period of time – there is a danger that other activities may arise that could divert resources and delay the rollout.
- Acquiring the necessary skills can be time consuming and costly.

4.1.9.2 Critical Success Factors

In order to obtain the necessary funding a compelling Business Case should be prepared showing how the benefits of effective Event Management can far outweigh the costs – giving a positive return on investment.

One of the most important CSFs is achieving the correct level of filtering. This is complicated by the fact that the significance of events changes. For example, a

user logging into a system today is normal, but if that user leaves the organization and tries to log in it is a security breach.

There are three keys to the correct level of filtering, as follows:

- Integrate Event Management into all Service Management processes where feasible. This will ensure that only the events significant to these processes are reported.
- Design new services with Event Management in mind (this is discussed in detail in paragraph 4.1.10).
- Trial and error. No matter how thoroughly Event Management is prepared, there will be classes of events that are not properly filtered. Event Management must therefore include a formal process to evaluate the effectiveness of filtering.

Proper planning is needed for the rollout of the monitoring agent software across the entire IT Infrastructure. This should be regarded as a project with realistic timescales and adequate resources being allocated and protected throughout the duration of the project.

4.1.9.3 Risks

The key risks are really those already mentioned above: failure to obtain adequate funding; ensuring the correct level of filtering; and failure to maintain momentum in rolling out the necessary monitoring agents across the IT Infrastructure. If any of these risks is not addressed it could adversely impact on the success of Event Management.

4.1.10 Designing for Event Management

Effective Event Management is not designed once a service has been deployed into Operations. Since Event Management is the basis for monitoring the performance and availability of a service, the exact targets and mechanisms for monitoring should be specified and agreed during the Availability and Capacity Management processes (see Service Design publication).

However, this does not mean that Event Management is designed by a group of remote system developers and then released to Operations Management together with the system that has to be managed. Nor does it mean that, once designed and agreed, Event Management becomes static – day-to-day operations will define additional events, priorities, alerts and other improvements that will feed through the Continual Improvement process back into Service Strategy, Service Design etc.

Service Operation functions will be expected to participate in the design of the service and how it is measured (see section 3.4).

For Event Management, the specific design areas include the following.

4.1.10.1 Instrumentation

Instrumentation is the definition of what can be monitored about CIs and the way in which their behaviour can be affected. In other words, instrumentation is about defining and designing exactly how to monitor and control the IT Infrastructure and IT services.

Instrumentation is partly about a set of decisions that need to be made and partly about designing mechanisms to execute these decisions.

Decisions that need to be made include:

- What needs to be monitored?
- What type of monitoring is required (e.g. active or passive; performance or output)?
- When do we need to generate an event?
- What type of information needs to be communicated in the event?
- Who are the messages intended for?

Mechanisms that need to be designed include:

- How will events be generated?
- Does the CI already have event generation mechanisms as a standard feature and, if so, which of these will be used? Are they sufficient or does the CI need to be customized to include additional mechanisms or information?
- What data will be used to populate the Event Record?
- Are events generated automatically or does the CI have to be polled?
- Where will events be logged and stored?
- How will supplementary data be gathered?

Note: A strong interface exists here with the application's design. All applications should be coded in such a way that meaningful and detailed error messages/codes are generated at the exact point of failure – so that these can be included in the event and allow swift diagnosis and resolution of the underlying cause. The need for the inclusion and testing of such error messaging is covered in more detail in the Service Transition publication.

4.1.10.2 Error messaging

Error messaging is important for all components (hardware, software, networks, etc.). It is particularly important that all software applications are designed to support Event Management. This might include the provision of meaningful error messages and/or codes that clearly indicate the specific point of failure and the

most likely cause. In such cases the testing of new applications should include testing of accurate event generation.

Newer technologies such as Java Management Extensions (JMX) or HawkNL™ provide the tools for building distributed, web-based, modular and dynamic solutions for managing and monitoring devices, applications and service-driven networks. These can be used to reduce or eliminate the need for programmers to include error messaging within the code – allowing a valuable level of normalization and code-independence.

4.1.10.3 Event Detection and Alert Mechanisms

Good Event Management design will also include the design and population of the tools used to filter, correlate and escalate Events.

The Correlation Engine specifically will need to be populated with the rules and criteria that will determine the significance and subsequent action for each type of event.

Thorough design of the event detection and alert mechanisms requires the following:

- Business knowledge in relationship to any business processes being managed via Event Management
- Detailed knowledge of the Service Level Requirements of the service being supported by each CI
- Knowledge of who is going to be supporting the CI
- Knowledge of what constitutes normal and abnormal operation of the CI
- Knowledge of the significance of multiple similar events (on the same CI or various similar CIs
- An understanding of what they need to know to support the CI effectively
- Information that can help in the diagnosis of problems with the CI
- Familiarity with incident prioritization and categorization codes so that if it is necessary to create an Incident Record, these codes can be provided
- Knowledge of other CIs that may be dependent on the affected CI, or those CIs on which it depends
- Availability of Known Error information from vendors or from previous experience.

4.1.10.4 Identification of thresholds

Thresholds themselves are not set and managed through Event Management. However, unless these are properly designed and communicated during the instrumentation process, it will be difficult to determine which level of performance is appropriate for each CI.

Also, most thresholds are not constant. They typically consist of a number of related variables. For example, the maximum number of concurrent users before response time slows will vary depending on what other jobs are active on the server. This knowledge is often only gained by experience, which means that Correlation Engines have to be continually tuned and updated through the process of Continual Service Improvement.

4.2 Incident Management

In ITIL terminology, an 'incident' is defined as:

An unplanned interruption to an IT service or reduction in the quality of an IT service. Failure of a configuration item that has not yet impacted service is also an incident, for example failure of one disk from a mirror set.

Incident Management is the process for dealing with all incidents; this can include failures, questions or queries reported by the users (usually via a telephone call to the Service Desk), by technical staff, or automatically detected and reported by event monitoring tools.

4.2.1 Purpose/goal/objective

The primary goal of the Incident Management process is to restore normal service operation as quickly as possible and minimize the adverse impact on business operations, thus ensuring that the best possible levels of service quality and availability are maintained. 'Normal service operation' is defined here as service operation within SLA limits.

4.2.2 Scope

Incident Management includes any event which disrupts, or which could disrupt, a service. This includes events which are communicated directly by users, either through the Service Desk or through an interface from Event Management to Incident Management tools.

Incidents can also be reported and/or logged by technical staff (if, for example, they notice something untoward with a hardware or network component they may report or log an incident and refer it to the Service Desk). This does not mean, however, that all events are incidents. Many classes of events are not related to disruptions at all, but are indicators of normal operation or are simply informational (see section 4.1).

Although both incidents and service requests are reported to the Service Desk, this does not mean that they are the same. Service requests do not represent a disruption to agreed service, but are a way of meeting the customer's needs and may be addressing an agreed target in an SLA. Service requests are dealt with by the Request Fulfilment process (see section 4.3).

4.2.3 Value to business

The value of Incident Management includes:

- The ability to detect and resolve incidents which results in lower downtime to the business, which in turn means higher availability of the service. This means that the business is able to exploit the functionality of the service as designed.
- The ability to align IT activity to real-time business priorities. This is because Incident Management includes the capability to identify business priorities and dynamically allocate resources as necessary.
- The ability to identify potential improvements to services. This happens as a result of understanding what constitutes an incident and also from being in contact with the activities of business operational staff.
- The Service Desk can, during its handling of incidents, identify additional service or training requirements found in IT or the business.

Incident Management is highly visible to the business, and it is therefore easier to demonstrate its value than most areas in Service Operation. For this reason, Incident Management is often one of the first processes to be implemented in Service Management projects. The added benefit of doing this is that Incident Management can be used to highlight other areas that need attention – thereby providing a justification for expenditure on implementing other processes.

4.2.4 Policies/principles/basic concepts

There are some basic things that need to be taken into account and decided when considering Incident Management. These are covered in this section.

4.2.4.1 Timescales

Timescales must be agreed for all incident-handling stages (these will differ depending upon the priority level of the incident) – based upon the overall incident response and resolution targets within SLAs – and captured as targets within OLAs and Underpinning Contracts (UCs). All support groups should be made fully aware of these timescales. Service Management tools should be used to automate timescales and escalate the incident as required based on pre-defined rules.

4.2.4.2 Incident Models

Many incidents are not new – they involve dealing with something that has happened before and may well happen again. For this reason, many organizations will find it helpful to pre-define ‘standard’ Incident Models – and apply them to appropriate incidents when they occur.

An Incident Model is a way of pre-defining the steps that should be taken to handle a process (in this case a process for dealing with a particular type of incident) in an agreed way. Support tools can then be used to manage the

required process. This will ensure that ‘standard’ incidents are handled in a pre-defined path and within pre-defined timescales.

Incidents which would require specialized handling can be treated in this way (for example, security-related incidents can be routed to Information Security Management and capacity- or performance-related incidents that would be routed to Capacity Management.

The Incident Model should include:

- The steps that should be taken to handle the incident
- 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
- Any necessary evidence-preservation activities (particularly relevant for security- and capacity-related incidents).

The models should be input to the incident-handling support tools in use and the tools should then automate the handling, management and escalation of the process.

4.2.4.3 Major incidents

A separate procedure, with shorter timescales and greater urgency, must be used for ‘major’ incidents. A definition of what constitutes a major incident must be agreed and ideally mapped on to the overall incident prioritization system – such that they will be dealt with through the major incident process.

Note: People sometimes use loose terminology and/or confuse a major incident with a problem. In reality, an incident remains an incident forever – it may grow in impact or priority to become a major incident, but an incident never ‘becomes’ a problem. A problem is the underlying cause of one or more incidents and remains a separate entity always!

Some lower-priority incidents may also have to be handled through this procedure – due to potential business impact – and some major incidents may not need to be handled in this way if the cause and resolutions are obvious and the normal incident process can easily cope within agreed target resolution times – provided the impact remains low!

Where necessary, the major incident procedure should include the dynamic establishment of a separate major incident team under the direct leadership of the Incident Manager, formulated to concentrate on this incident alone to ensure that adequate resources and focus are provided to finding a swift resolution. If

the Service Desk Manager is also fulfilling the role of Incident Manager (say in a small organization), then a separate person may need to be designated to lead the major incident investigation team – so as to avoid conflict of time or priorities – but should ultimately report back to the Incident Manager.

If the cause of the incident needs to be investigated at the same time, then the Problem Manager would be involved as well but the Incident Manager must ensure that service restoration and underlying cause are kept separate. Throughout, the Service Desk would ensure that all activities are recorded and users are kept fully informed of progress.

4.2.5 Process activities, methods and techniques

The process to be followed during the management of an incident is shown in Figure 4.2. The process includes the following steps.

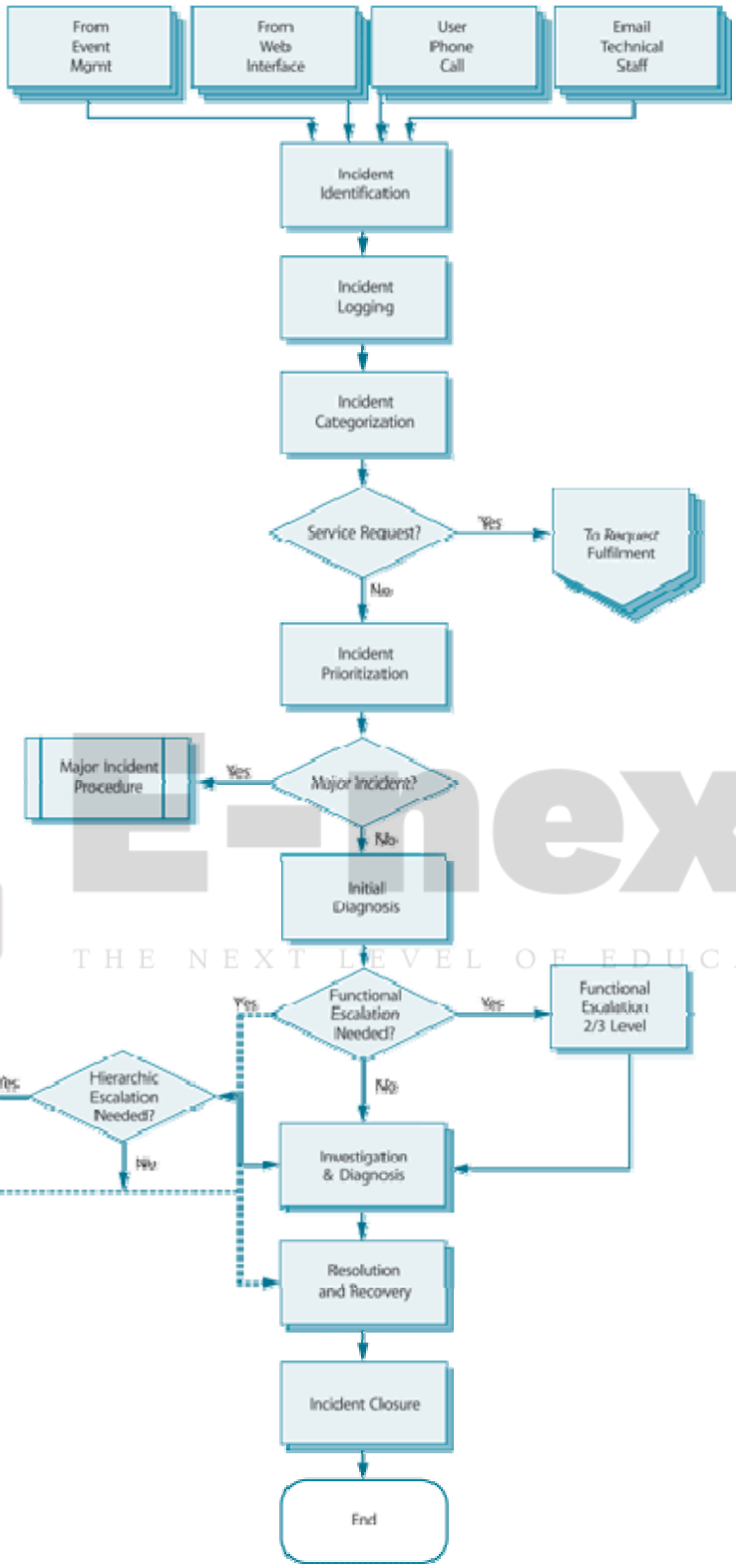

Figure 4.2 Incident Management process flow

4.2.5.1 Incident identification

Work cannot begin on dealing with an incident until it is known that an incident has occurred. It is usually unacceptable, from a business perspective, to wait until a user is impacted and contacts the Service Desk. As far as possible, all key components should be monitored so that failures or potential failures are detected early so that the Incident Management process can be started quickly. Ideally, incidents should be resolved before they have an impact on users!

Please see section 4.1 for further details.

4.2.5.2 Incident logging

All incidents must be fully logged and date/time stamped, regardless of whether they are raised through a Service Desk telephone call or whether automatically detected via an event alert.

Note: If Service Desk and/or support staff visit the customers to deal with one incident, they may be asked to deal with further incidents 'while they are there'. It is important that if this is done, a separate Incident Record is logged for each additional incident handled – to ensure that a historical record is kept and credit is given for the work undertaken.

All relevant information relating to the nature of the incident must be logged so that a full historical record is maintained – and so that if the incident has to be referred to other support group(s), they will have all relevant information to hand to assist them.

The information needed for each incident is likely to include:

- Unique reference number
- Incident categorization (often broken down into between two and four levels of sub-categories)
- Incident urgency
- Incident impact
- Incident prioritization
- Date/time recorded
- Name/ID of the person and/or group recording the incident
- Method of notification (telephone, automatic, e-mail, in person, etc.)
- Name/department/phone/location of user
- Call-back method (telephone, mail, etc.)
- Description of symptoms
- Incident status (active, waiting, closed, etc.)
- Related CI
- Support group/person to which the incident is allocated
- Related problem/Known Error
- Activities undertaken to resolve the incident
- Resolution date and time

- Closure category
- Closure date and time.

Note: If the Service Desk does not work 24/7 and responsibility for first-line incident logging and handling passes to another group, such as IT Operations or Network Support, out of Service Desk hours, then these staff need to be equally rigorous about logging of incident details. Full training and awareness needs to be provided to such staff on this issue.

4.2.5.3 Incident categorization

Part of the initial logging must be to allocate suitable incident categorization coding so that the exact type of the call is recorded. This will be important later when looking at incident types/frequencies to establish trends for use in Problem Management, Supplier Management and other ITSM activities.

Please note that the check for Service Requests in this process does not imply that Service Requests are incidents. This is simply recognition of the fact that Service Requests are sometimes incorrectly logged as incidents (e.g. a user incorrectly enters the request as an incident from the web interface). This check will detect any such requests and ensure that they are passed to the Request Fulfilment process.

Multi-level categorization is available in most tools – usually to three or four levels of granularity. For example, an incident may be categorized as shown in Figure 4.3.

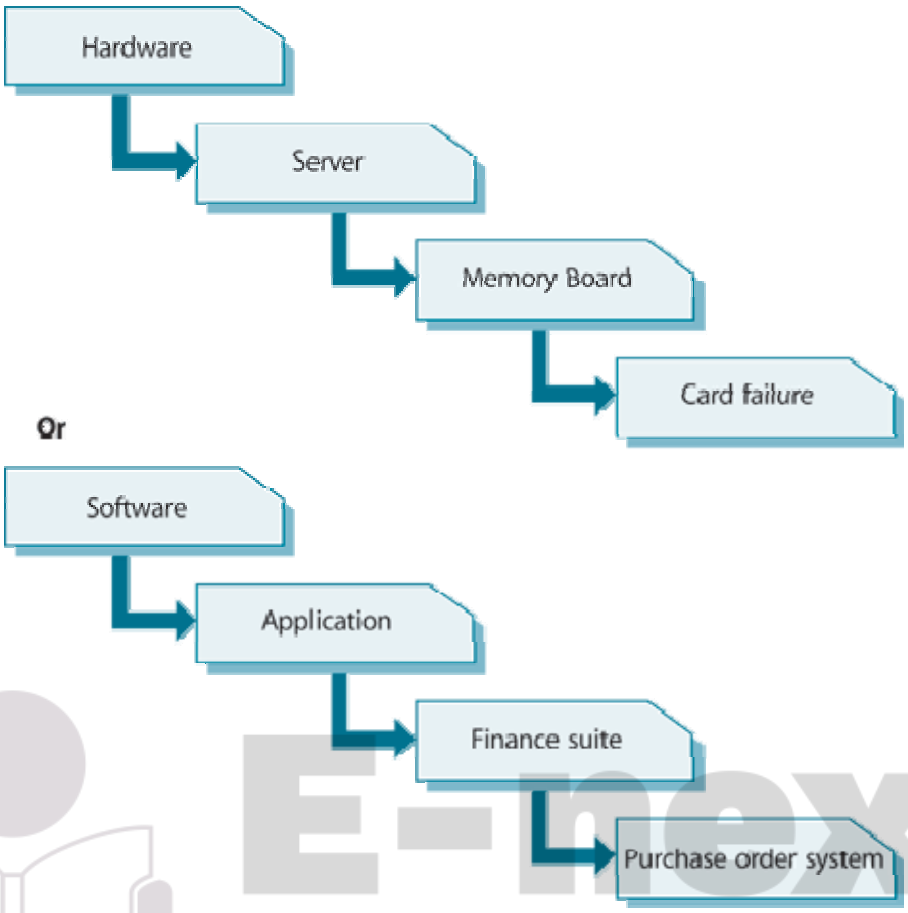

Figure 4.3 Multi-level incident categorization

All organizations are unique and it is therefore difficult to give generic guidance on the categories an organization should use, particularly at the lower levels. However, there is a technique that can be used to assist an organization to achieve a correct and complete set of categories – if they are starting from scratch! The steps involve:

1. Hold a brainstorming session among the relevant support groups, involving the SD Supervisor and Incident and Problem Managers.
2. Use this session to decide the 'best guess' top-level categories – and include an 'other' category. Set up the relevant logging tools to use these categories for a trial period.
3. Use the categories for a short trial period (long enough for several hundred incidents to fall into each category, but not too long that an analysis will take too long to perform).
4. Perform an analysis of the incidents logged during the trial period. The number of incidents logged in each higher-level category will confirm whether the categories are worth having – and a more detailed analysis of the 'other' category should allow identification of any additional higher-level categories that will be needed.

5. A breakdown analysis of the incidents within each higher-level category should be used to decide the lower-level categories that will be required.
6. Review and repeat these activities after a further period – of, say, one to three months – and again regularly to ensure that they remain relevant. Be aware that any significant changes to categorization may cause some difficulties for incident trending or management reporting – so they should be stabilized unless changes are genuinely required.

If an existing categorization scheme is in use, but it is not thought to be working satisfactorily, the basic idea of the technique suggested above can be used to review and amend the existing scheme.

NOTE: Sometimes the details available at the time an incident is logged may be incomplete, misleading or incorrect. It is therefore important that the categorization of the incident is checked, and updated if necessary, at call closure time (in a separate closure categorization field, so as not to corrupt the original categorization) – please see paragraph 4.2.5.9.

4.2.5.4 Incident prioritization

Another important aspect of logging every incident is to agree and allocate an appropriate prioritization code – as this will determine how the incident is handled both by support tools and support staff.

Prioritization can normally be determined by taking into account both the urgency of the incident (how quickly the business needs a resolution) and the level of impact it is causing. An indication of impact is often (but not always) the number of users being affected. In some cases, and very importantly, the loss of service to a single user can have a major business impact – it all depends upon who is trying to do what – so numbers alone is not enough to evaluate overall priority! Other factors that can also contribute to impact levels are:

- Risk to life or limb
- The number of services affected – may be multiple services
- The level of financial losses
- Effect on business reputation
- Regulatory or legislative breaches.

An effective way of calculating these elements and deriving an overall priority level for each incident is given in Table 4.1:

			Impact	
		High	Medium	Low
	High	1	2	3
Urgency	Medium	2	3	4
	Low	3	4	5

Priority code	Description	Target resolution time
1	Critical	1 hour
2	High	8 hours
3	Medium	24 hours
4	Low	48 hours
5	Planning	Planned

Table 4.1 Simple priority coding system

In all cases, clear guidance – with practical examples – should be provided for all support staff to enable them to determine the correct urgency and impact levels, so the correct priority is allocated. Such guidance should be produced during service level negotiations.

However, it must be noted that there will be occasions when, because of particular business expediency or whatever, normal priority levels have to be overridden. When a user is adamant that an incident's priority level should exceed normal guidelines, the Service Desk should comply with such a request – and if it subsequently turns out to be incorrect this can be resolved as an off-line management level issue, rather than a dispute occurring when the user is on the telephone.

Some organizations may also recognize VIPs (high-ranking executives, officers, diplomats, politicians, etc.) whose incidents would be handled on a higher priority than normal – but in such cases this is best catered for and documented within the guidance provided to the Service Desk staff on how to apply the priority levels, so they are all aware of the agreed rules for VIPs, and who falls into this category.

It should be noted that an incident's priority may be dynamic – if circumstances change, or if an incident is not resolved within SLA target times, then the priority must be altered to reflect the new situation.

Note: some tools may have constraints that make it difficult automatically to calculate performance against SLA targets if a priority is changed during the lifetime of an incident. However, if circumstances do change, the change in priority should be made – and if necessary manual adjustments made to reporting tools. Ideally, tools with such constraints should not be selected.

4.2.5.5 Initial diagnosis

If the incident has been routed via the Service Desk, the Service Desk Analyst must carry out initial diagnosis, typically while the user is still on the telephone – if the call is raised in this way – to try to discover the full symptoms of the incident and to determine exactly what has gone wrong and how to correct it. It is at this stage that diagnostic scripts and known error information can be most valuable in allowing earlier and accurate diagnosis.

If possible, the Service Desk Analyst will resolve the incident while the user is still on the telephone – and close the incident if the resolution is successful.

If the Service Desk Analyst cannot resolve the incident while the user is still on the telephone, but there is a prospect that the Service Desk may be able to do so within the agreed time limit without assistance from other support groups, the Analyst should inform the user of their intentions, give the user the incident reference number and attempt to find a resolution.

4.2.5.6 Incident escalation

- Functional escalation. As soon as it becomes clear that the Service Desk is unable to resolve the incident itself (or when target times for first-point resolution have been exceeded – whichever comes first!) the incident must be immediately escalated for further support.

  If the organization has a second-level support group and the Service Desk believes that the incident can be resolved by that group, it should refer the incident to them. If it is obvious that the incident will need deeper technical knowledge – or when the second-level group has not been able to resolve the incident within agreed target times (whichever comes first), the incident must be immediately escalated to the appropriate third-level support group. Note that third-level support groups may be internal – but they may also be third parties such as software suppliers or hardware manufacturers or maintainers. The rules for escalation and handling of incidents must be agreed in OLAs and UCs with internal and external support groups respectively.

  Note: Incident Ownership remains with the Service Desk! Regardless of where an incident is referred to during its life, ownership of the incident remains with the Service Desk at all times. The Service Desk remains

responsible for tracking progress, keeping users informed and ultimately for Incident Closure.

- Hierarchic escalation. If incidents are of a serious nature (for example Priority 1 incidents) the appropriate IT managers must be notified, for informational purposes at least. Hierarchic escalation is also used if the 'Investigation and Diagnosis' and 'Resolution and Recovery' steps are taking too long or proving too difficult. Hierarchic escalation should continue up the management chain so that senior managers are aware and can be prepared and take any necessary action, such as allocating additional resources or involving suppliers/maintainers. Hierarchic escalation is also used when there is contention about to whom the incident is allocated.

  Hierarchic escalation can, of course, be initiated by the affected users or customer management, as they see fit – that is why it is important that IT managers are made aware so that they can anticipate and prepare for any such escalation.

The exact levels and timescales for both functional and hierarchic escalation need to be agreed, taking into account SLA targets, and embedded within support tools which can then be used to police and control the process flow within agreed timescales.

The Service Desk should keep the user informed of any relevant escalation that takes place and ensure the Incident Record is updated accordingly to keep a full history of actions.

Note regarding Incident allocation

There may be many incidents in a queue with the same priority level – so it will be the job of the Service Desk and/or Incident Management staff initially, in conjunction with managers of the various support groups to which incidents are escalated, to decide the order in which incidents should be picked up and actively worked on. These managers must ensure that incidents are dealt with in true business priority order and that staff are not allowed to 'cherry-pick' the incidents they choose!

4.2.5.7 Investigation and Diagnosis

In the case of incidents where the user is just seeking information, the Service Desk should be able to provide this fairly quickly and resolve the service request – but if a fault is being reported, this is an incident and likely to require some degree of investigation and diagnosis.

Each of the support groups involved with the incident handling will investigate and diagnose what has gone wrong – and all such activities (including details of any actions taken to try to resolve or re-create the incident) should be fully documented in the incident record so that a complete historical record of all activities is maintained at all times.

Note: Valuable time can often be lost if investigation and diagnostic action (or indeed resolution or recovery actions) are performed serially. Where possible, such activities should be performed in parallel to reduce overall timescales – and support tools should be designed and/or selected to allow this. However, care should be taken to coordinate activities, particularly resolution or recovery activities, otherwise the actions of different groups may conflict or further complicate a resolution!

This investigation is likely to include such actions as:

- Establishing exactly what has gone wrong or being sought by the user
- Understanding the chronological order of events
- Confirming the full impact of the incident, including the number and range of users affected
- Identifying any events that could have triggered the incident (e.g. a recent change, some user action?)
- Knowledge searches looking for previous occurrences by searching previous Incident/Problem Records and/or Known Error Databases or manufacturers'/suppliers' Error Logs or Knowledge Databases.

4.2.5.8 Resolution and Recovery

When a potential resolution has been identified, this should be applied and tested. The specific actions to be undertaken and the people who will be involved in taking the recovery actions may vary, depending upon the nature of the fault – but could involve:

- Asking the user to undertake directed activities on their own desk top or remote equipment
- The Service Desk implementing the resolution either centrally (say, rebooting a server) or remotely using software to take control of the user's desktop to diagnose and implement a resolution
- Specialist support groups being asked to implement specific recovery actions (e.g. Network Support reconfiguring a router)
- A third-party supplier or maintainer being asked to resolve the fault.

Even when a resolution has been found, sufficient testing must be performed to ensure that recovery action is complete and that the service has been fully restored to the user(s).

NOTE: in some cases it may be necessary for two or more groups to take separate, though perhaps coordinated, recovery actions for an overall resolution to be implemented. In such cases Incident Management must coordinate the activities and liaise with all parties involved.

Regardless of the actions taken, or who does them, the Incident Record must be updated accordingly with all relevant information and details so that a full history is maintained.

The resolving group should pass the incident back to the Service Desk for closure action.

4.2.5.9 Incident Closure

The Service Desk should check that the incident is fully resolved and that the users are satisfied and willing to agree the incident can be closed. The Service Desk should also check the following:

- **Closure categorization.** Check and confirm that the initial incident categorization was correct or, where the categorization subsequently turned out to be incorrect, update the record so that a correct closure categorization is recorded for the incident – seeking advise or guidance from the resolving group(s) as necessary.
- **User satisfaction survey.** Carry out a user satisfaction call-back or e-mail survey for the agreed percentage of incidents.
- **Incident documentation.** Chase any outstanding details and ensure that the Incident Record is fully documented so that a full historic record at a sufficient level of detail is complete.
- **Ongoing or recurring** problem**?** Determine (in conjunction with resolver groups) whether it is likely that the incident could recur and decide whether any preventive action is necessary to avoid this. In conjunction with Problem Management, raise a Problem Record in all such cases so that preventive action is initiated.
- **Formal** closure**.** Formally close the Incident Record.

Note: Some organizations may chose to utilize an automatic closure period on specific, or even all, incidents (e.g. incident will be automatically closed after two working days if no further contact is made by the user). Where this approach is to be considered, it must first be fully discussed and agreed with the users – and widely publicized so that all users and IT staff are aware of this. It may be inappropriate to use this method for certain types of incidents – such as major incidents or those involving VIPs, etc.

Rules for re-opening incidents

Despite all adequate care, there will be occasions when incidents recur even though they have been formally closed. Because of such cases, it is wise to have pre-defined rules about if and when an incident can be re-opened. It might make sense, for example, to agree that if the incident recurs within one working day then it can be re-opened – but that beyond this point a new incident must be raised, but linked to the previous incident(s).

The exact time threshold/rules may vary between individual organizations – but clear rules should be agreed and documented and guidance given to all Service Desk staff so that uniformity is applied.

4.2.6 Triggers, input and output/inter-process interfaces

Incidents can be triggered in many ways. The most common route is when a user rings the Service Desk or completes a web-based incident-logging screen, but increasingly incidents are raised automatically via Event Management tools. Technical staff may notice potential failures and raise an incident, or ask the Service Desk to do so, so that the fault can be addressed. Some incidents may also arise at the initiation of suppliers – who may send some form of notification of a potential or actual difficulty that needs attention.

The interfaces with Incident Management include:

- Problem Management**:** Incident Management forms part of the overall process of dealing with problems in the organization. Incidents are often caused by underlying problems, which must be solved to prevent the incident from recurring. Incident Management provides a point where these are reported.
- Configuration Management provides the data used to identify and progress incidents. One of the uses of the CMS is to identify faulty equipment and to assess the impact of an incident. It is also used to identify the users affected by potential problems. The CMS also contains information about which categories of incident should be assigned to which support group. In turn, Incident Management can maintain the status of faulty CIs. It can also assist Configuration Management to audit the infrastructure when working to resolve an incident.
- Change Management**:** Where a change is required to implement a workaround or resolution, this will need to be logged as an RFC and progressed through Change Management. In turn, Incident Management is able to detect and resolve incidents that arise from failed changes.
- Capacity Management**:** Incident Management provides a trigger for performance monitoring where there appears to be a performance problem. Capacity Management may develop workarounds for incidents.

- Availability Management**;** will use Incident Management data to determine the availability of IT services and look at where the incident lifecycle can be improved.
- **SLM:** The ability to resolve incidents in a specified time is a key part of delivering an agreed level of service. Incident Management enables SLM to define measurable responses to service disruptions. It also provides reports that enable SLM to review SLAs objectively and regularly. In particular, Incident Management is able to assist in defining where services are at their weakest, so that SLM can define actions as part of the Service Improvement Programme (SIP) – please see the Continual Service Improvement publication for more details. SLM defines the acceptable levels of service within which Incident Management works, including:
    - Incident response times
    - Impact definitions
    - Target fix times
    - Service definitions, which are mapped to users
    - Rules for requesting services
    - Expectations for providing feedback to users.

4.2.7 Information Management

Most information used in Incident Management comes from the following sources:

- **The Incident Management tools**, which contain information about:
    - Incident and problem history
    - Incident categories
    - Action taken to resolve incidents
    - Diagnostic scripts which can help first-line analysts to resolve the incident, or at least gather information that will help second- or third-line analysts resolve it faster.
- Incident Record**s**, which include the following data:
    - Unique reference number
    - Incident classification
    - Date and time of recording and any subsequent activities
    - Name and identity of the person recording and updating the Incident Record
    - Name/organization/contact details of affected user(s)
    - Description of the incident symptoms
    - Details of any actions taken to try to diagnose, resolve or re-create the incident
    - Incident category, impact, urgency and priority
    - Relationship with other incidents, problems, changes or Known Errors

- Closure details, including time, category, action taken and identity of person closing the record.

Incident Management also requires access to the CMS. This will help it to identify the CIs affected by the incident and also to estimate the impact of the incident.

The Known Error Database provides valuable information about possible resolutions and workarounds. This is discussed in detail in paragraph 4.4.7.2.

4.2.8 Metrics

The metrics that should be monitored and reported upon to judge the efficiency and effectiveness of the Incident Management process, and its operation, will include:

- Total numbers of Incidents (as a control measure)
- Breakdown of incidents at each stage (e.g. logged, work in progress, closed etc)
- Size of current incident backlog
- Number and percentage of major incidents
- Mean elapsed time to achieve incident resolution or circumvention, broken down by impact code
- Percentage of incidents handled within agreed response time (incident response-time targets may be specified in SLAs, for example, by impact and urgency codes)
- Average cost per incident
- Number of incidents reopened and as a percentage of the total
- Number and percentage of incidents incorrectly assigned
- Number and percentage of incidents incorrectly categorized
- Percentage of Incidents closed by the Service Desk without reference to other levels of support (often referred to as ‘first point of contact’)
- Number and percentage the of incidents processed per Service Desk agent
- Number and percentage of incidents resolved remotely, without the need for a visit
- Number of incidents handled by each Incident Model
- Breakdown of incidents by time of day, to help pinpoint peaks and ensure matching of resources.

Reports should be produced under the authority of the Incident Manager, who should draw up a schedule and distribution list, in collaboration with the Service Desk and support groups handling incidents. Distribution lists should at least include IT services Management and specialist support groups. Consider also making the data available to users and customers, for example via SLA reports.

4.2.9 Challenges, Critical Success Factors and risks

4.2.9.1 Challenges

The following challenges will exist for successful Incident Management:

- The ability to detect incidents as early as possible. This will require education of the users reporting incidents, the use of Super Users (see paragraph 6.2.4.5) and the configuration of Event Management tools.
- Convincing all staff (technical teams as well as users) that all incidents must be logged, and encouraging the use of self-help web-based capabilities (which can speed up assistance and reduce resource requirements).
- Availability of information about problems and Known Errors. This will enable Incident Management staff to learn from previous incidents and also to track the status of resolutions.
- Integration into the CMS to determine relationships between CIs and to refer to the history of CIs when performing first-line support.
- Integration into the SLM process. This will assist Incident Management correctly to assess the impact and priority of incidents and assists in defining and executing escalation procedures. SLM will also benefit from the information learned during Incident Management, for example in determining whether service level performance targets are realistic and achievable.

4.2.9.2 Critical Success Factors

The following factors will be critical for successful Incident Management:

- A good Service Desk is key to successful Incident Management
- Clearly defined targets to work to – as defined in SLAs
- Adequate customer-oriented and technically training support staff with the correct skill levels, at all stages of the process
- Integrated support tools to drive and control the process
- OLAs and UCs that are capable of influencing and shaping the correct behaviour of all support staff.

4.2.9.3 Risks

The risks to successful Incident Management are actually similar to some of the challenges and the reverse of some of the Critical Success Factors mentioned above. They include:

- Being inundated with incidents that cannot be handled within acceptable timescales due to a lack of available or properly trained resources

- Incidents being bogged down and not progressed as intended because of inadequate support tools to raise alerts and prompt progress
- Lack of adequate and/or timely information sources because of inadequate tools or lack of integration
- Mismatches in objectives or actions because of poorly aligned or non-existent OLAs and/or UCs.

4.3 Request Fulfilment

The term 'Service Request' is used as a generic description for many varying types of demands that are placed upon the IT Department by the users. Many of these are actually small changes – low risk, frequently occurring, low cost, etc. (e.g. a request to change a password, a request to install an additional software application onto a particular workstation, a request to relocate some items of desktop equipment) or maybe just a question requesting information – but their scale and frequent, low-risk nature means that they are better handled by a separate process, rather than being allowed to congest and obstruct the normal Incident and Change Management processes.

4.3.1 Purpose/goal/objective

Request Fulfilment is the processes of dealing with Service Requests from the users. The objectives of the Request Fulfilment process include:

- To provide a channel for users to request and receive standard services for which a pre-defined approval and qualification process exists
- To provide information to users and customers about the availability of services and the procedure for obtaining them
- To source and deliver the components of requested standard services (e.g. licences and software media)
- To assist with general information, complaints or comments.

4.3.2 Scope

The process needed to fulfil a request will vary depending upon exactly what is being requested – but can usually be broken down into a set of activities that have to be performed. Some organizations will be comfortable to let the Service Requests be handled through their Incident Management processes (and tools) – with Service Requests being handled as a particular type of 'incident' (using a high-level categorization system to identify those 'incidents' that are in fact Service Requests).

Note, however, that there is a significant difference here – an incident is usually an unplanned event whereas a Service Request is usually something that can and should be planned!

Therefore, in an organization where large numbers of Service Requests have to be handled, and where the actions to be taken to fulfil those requests are very varied or specialized, it may be appropriate to handle Service Requests as a completely separate work stream – and to record and manage them as a separate record type.

This may be particularly appropriate if the organization has chosen to widen the scope of the Service Desk to expand upon just IT-related issues and use the desk as a focal point for other types or request for service – for example, a request to service a photocopier or even going so far as to include, for example, building management issues, such as a need to replace a light fitment or repair a leak in the plumbing.

Note: It will ultimately be up to each organization to decide and document which request it will handle through the Request Fulfilment process and which others will have to go through more formal Change Management. There will always be grey areas which prevent generic guidance from being usefully prescribed.

4.3.3 Value to business

The value of Request Fulfilment is to provide quick and effective access to standard services which business staff can use to improve their productivity or the quality of business services and products.

Request Fulfilment effectively reduces the bureaucracy involved in requesting and receiving access to existing or new services, thus also reducing the cost of providing these services. Centralizing fulfilment also increases the level of control over these services. This in turn can help reduce costs through centralized negotiation with suppliers, and can also help to reduce the cost of support.

4.3.4 Policies/principles/basic concepts

Many Service Requests will be frequently recurring, so a predefined process flow (a model) can be devised to include the stages needed to fulfil the request, the individuals or support groups involved, target timescales and escalation paths. Service Requests will usually be satisfied by implementing a Standard Change (see the Service Transition publication for further details on Standard Changes). The ownership of Service Requests resides with the Service Desk, which monitors, escalates, dispatches and often fulfils the user request.

4.3.4.1 Request Models

Some Service Requests will occur frequently and will require handling in a consistent manner in order to meet agreed service levels. To assist this, many organizations will wish to create pre-defined Request Models (which typically include some form of pre-approval by Change Management). This is similar in concept to the idea of Incident Models already described in paragraph 4.2.4.2, but applied to Service Requests.

4.3.5 Process activities, methods and techniques

4.3.5.1 Menu selection

Request Fulfilment offers great opportunities for self-help practices where users can generate a Service Request using technology that links into Service Management tools. Ideally, users should be offered a 'menu'-type selection via a web interface, so that they can select and input details of Service Requests from a pre-defined list –where appropriate expectations can be set by giving target delivery and/or implementation targets/dates (in line with SLA targets). Where organizations are offering a self-help IT support capability to the users, it would make sense to combine this with a Request Fulfilment system as described.

Specialist web tools to offer this type of 'shopping basket' experience can be used together with interfaces directly to the back-end integrated ITSM tools, or other more general business process automation or Enterprise Resource Planning (ERP) tools that may be used for management of the Request Fulfilment activities.

4.3.5.2 Financial approval

One important extra step that is likely to be needed when dealing with a service request is that of financial approval.

Most requests will have some form of financial implications, regardless of the type of commercial arrangements in place. The cost of fulfilling the request must first be established. It may be possible to agree fixed prices for 'standard' requests – and prior approval for such requests may be given as part of the organization's overall annual financial management. In all other cases, an estimate of the cost must be produced and submitted to the user for financial approval (the user may need to seek approval up their management/financial chain). If approval is given, in addition to fulfilling the request, the process must also include charging (billing or cross-charging) for the work done – if charging is in place.

4.3.5.3 Other approval

In some cases further approval may be needed – such as compliance-related or wider business approval. Request Fulfilment must have the ability to define and check such approvals where needed.

4.3.5.4 Fulfilment

The actual fulfilment activity will depend upon the nature of the Service Request. Some simpler requests may be completed by the Service Desk, acting as first-line support, while others will have to be forwarded to specialist groups and/or suppliers for fulfilment.

Some organizations may have specialist fulfilment groups (to 'pick, pack and dispatch') – or may have outsourced some fulfilment activities to a third-party

supplier(s). The Service Desk should monitor and chase progress and keep users informed throughout, regardless of the actual fulfilment source.

4.3.5.5 Closure

When the Service Request has been fulfilled it must be referred back to the Service Desk for closure. The Service Desk should go through the same closure process as described earlier in paragraph 4.2.5.9 – checking that the user is satisfied with the outcome.

4.3.6 Triggers, input and output/inter-process interfaces

Most requests will be triggered through either a user calling the Service Desk or a user completing some form of self-help web-based input screen to make their request. The latter will often involve a selection from a portfolio of available request types.The primary interfaces with Request Fulfilment include:

- **Service Desk/**Incident Management**:** Many Service Requests may come in via the Service Desk and may be initially handled through the Incident Management process. Some organizations may choose that all requests are handled via this route – but others may choose to have a separate process, for reasons already discussed earlier in this chapter.
- A strong link is also needed between **Request Fulfilment,** Release**, Asset and** Configuration Management – as some requests will be for the deployment of new or upgraded components that can be automatically deployed. In such cases the 'release' can be pre-defined, built and tested but only deployed upon request by those who want the 'release'. Upon deployment, the CMS will have to be updated to reflect the change. Where appropriate, software licence checks/updates will also be necessary.

Where appropriate, it will be necessary to relate IT-related Service Requests to any incidents or problems that have initiated the need for the request (as would be the case for any other type of change).

4.3.7 Information Management

Request Fulfilment is dependent on information from the following sources:

- The Service Requests will contain information about:
  - What service is being requested
  - Who requested and authorized the service
  - Which process will be used to fulfil the request
  - To whom it was assigned to and what action was taken
  - The date and time when the request was logged as well as the date and time of all actions taken
  - Closure details.

- Requests for Change: In some cases the Request Fulfilment process will be initiated by an RFC. This is typical where the Service Request relates to a CI
- The Service Portfolio, to enable the scope of agreed Service Request to be identified
- Security Policies will prescribe any controls to be executed or adhered to when providing the service, e.g. ensuring that the requester is authorized to access the service, or that the software is licensed.

4.3.8 Metrics

The metrics needed to judge the effectiveness and efficiency of Request Fulfilment will include the following (each metric will need to be broken down by request type, within the period):

- The total number of Service Requests (as a control measure)
- Breakdown of service requests at each stage (e.g. logged, WIP, closed, etc.)
- The size of current backlog of outstanding Service Requests
- The mean elapsed time for handling each type of Service Request
- The number and percentage of Service Requests completed within agreed target times
- The average cost per type of Service Request
- Level of client satisfaction with the handling of Service Requests (as measured in some form of satisfaction survey).

4.3.9 Challenges, Critical Success Factors and risks

4.3.9.1 Challenges

The following challenges will be faced when introducing Request Fulfilment:

- Clearly defining and documenting the type of requests that will be handled within the Request Fulfilment process (and those that will either go through the Service Desk and be handled as incidents or those that will need to go through formal Change Management) – so that all parties are absolutely clear on the scope.
- Establishing self-help front-end capabilities that allow the users to interface successfully with the Request Fulfilment process.

4.3.9.2 Critical Success Factors

Request Fulfilment depends on the following Critical Success Factors:

- Agreement of what services will be standardized and who is authorized to request them. The cost of these services must also be agreed. This may

be done as part of the SLM process. Any variances of the services must also be defined.

- Publication of the services to users as part of the Service Catalogue. It is important that this part of the Service Catalogue must be easily accessed, perhaps on the Intranet, and should be recognized as the first source of information for users seeking access to a service.
- Definition of a standard fulfilment procedure for each of the services being requested. This includes all procurement policies and the ability to generate purchase orders and work orders
- A single point of contact which can be used to request the service. This is often provided by the Service Desk or through an Intranet request, but could be through an automated request directly into the Request Fulfilment or procurement system.
- Self-service tools needed to provide a front-end interface to the users. It is essential that these integrate with the back-end fulfilment tools, often managed through Incident or Change Management.

4.3.9.3 Risks

Risks that may be encountered with Request Fulfilment include:

- Poorly defined scope, where people are unclear about exactly what the process is expected to handle
- Poorly designed or implemented user interfaces so that users have difficulty raising the requests that they need
- Badly designed or operated back-end fulfilment processes that are incapable of dealing with the volume or nature of the requests being made
- Inadequate monitoring capabilities so that accurate metrics cannot be gathered.

4.4 Problem Management

ITIL defines a 'problem' as the cause of one or more incidents.

4.4.1 Purpose/goal/objective

Problem Management is the process responsible for managing the lifecycle of all problems. The primary objectives of Problem Management are to prevent problems and resulting incidents from happening, to eliminate recurring incidents and to minimize the impact of incidents that cannot be prevented.

4.4.2 Scope

Problem Management includes the activities required to diagnose the root cause of incidents and to determine the resolution to those problems. It is also responsible for ensuring that the resolution is implemented through the appropriate control procedures, especially Change Management and Release Management.

Problem Management will also maintain information about problems and the appropriate workarounds and resolutions, so that the organization is able to reduce the number and impact of incidents over time. In this respect, Problem Management has a strong interface with Knowledge Management, and tools such as the Known Error Database will be used for both.

Although Incident and Problem Management are separate processes, they are closely related and will typically use the same tools, and may use similar categorization, impact and priority coding systems. This will ensure effective communication when dealing with related incidents and problems.

4.4.3 Value to business

Problem Management works together with Incident Management and Change Management to ensure that IT service availability and quality are increased. When incidents are resolved, information about the resolution is recorded. Over time, this information is used to speed up the resolution time and identify permanent solutions, reducing the number and resolution time of incidents. This results in less downtime and less disruption to business critical systems.

Additional value is derived from the following:

- Higher availability of IT services
- Higher productivity of business and IT staff
- Reduced expenditure on workarounds or fixes that do not work
- Reduction in cost of effort in fire-fighting or resolving repeat incidents.

4.4.4 Policies/principles/basic concepts

There are some important concepts of Problem Management that must be taken into account from the outset. These include:

4.4.4.1 Problem Models

Many problems will be unique and will require handling in an individual way – but it is conceivable that some incidents may recur because of dormant or underlying problems (for example, where the cost of a permanent resolution will be high and a decision has been taken not to go ahead with an expensive solution – but to 'live with' the problem).

As well as creating a Known Error Record in the Known Error Database (see paragraph 4.4.5.7) to ensure quicker diagnosis, the creation of a Problem Model for handling such problems in the future may be helpful. This is very similar in concept to the idea of Incident Models already described in paragraph 4.2.4.2, but applied to problems as well as incidents.

4.4.5 Process activities, methods and techniques

Problem Management consists of two major processes:

- **Reactive** Problem Management, which is generally executed as part of Service Operation – and is therefore covered in this publication
- Proactive Problem Management which is initiated in Service Operation, but generally driven as part of Continual Service Improvement.

The reactive Problem Management process is shown in Figure 4.4. This is a simplified chart to show the normal process flow, but in reality some of the states may be iterative or variations may have to be made in order to handle particular situations.

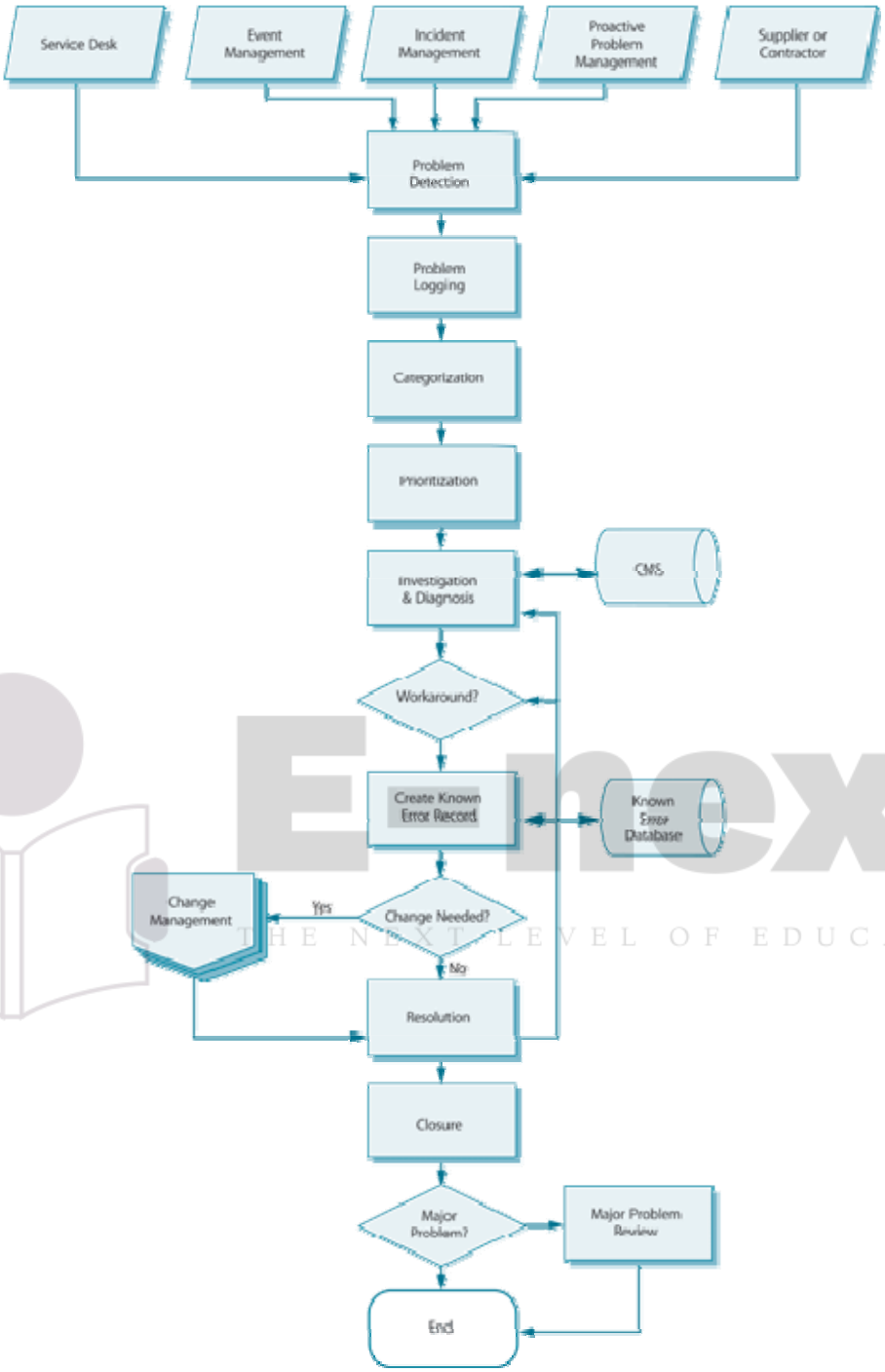

Figure 4.4 Problem Management process flow

4.4.5.1 Problem detection

It is likely that multiple ways of detecting problems will exist in all organizations. These will include:

- Suspicion or detection of a cause of one or more incidents by the Service Desk, resulting in a Problem Record being raised – the desk may have resolved the incident but has not determined a definitive cause and suspects that it is likely to recur, so will raise a Problem Record to allow the underlying cause to be resolved. Alternatively, it may be immediately obvious from the outset that an incident, or incidents, has been caused by a major problem, so a Problem Record will be raised without delay.
- Analysis of an incident by a technical support group which reveals that an underlying problem exists, or is likely to exist.
- Automated detection of an infrastructure or application fault, using event/alert tools automatically to raise an incident which may reveal the need for a Problem Record.
- A notification from a supplier or contractor that a problem exists that has to be resolved.
- Analysis of incidents as part of proactive Problem Management – resulting in the need to raise a Problem Record so that the underlying fault can be investigated further.

Frequent and regular analysis of incident and problem data must be performed to identify any trends as they become discernible. This will require meaningful and detailed categorization of incidents/problems and regular reporting of patterns and areas of high occurrence. 'Top ten' reporting, with drill-down capabilities to lower levels, is useful in identifying trends.

Further details of how detected trends should be handled are included in the Continual Service Improvement publication.

4.4.5.2 Problem logging

Regardless of the detection method, all the relevant details of the problem must be recorded so that a full historic record exists. This must be date and time stamped to allow suitable control and escalation.

A cross-reference must be made to the incident(s) which initiated the Problem Record – and all relevant details must be copied from the Incident Record(s) to the Problem Record. It is difficult to be exact, as cases may vary, but typically this will include details such as:

- User details
- Service details
- Equipment details
- Date/time initially logged
- Priority and categorization details
- Incident description
- Details of all diagnostic or attempted recovery actions taken.

4.4.5.3 Problem Categorization

Problems must be categorized in the same way as incidents (and it is advisable to use the same coding system) so that the true nature of the problem can be easily traced in the future and meaningful management information can be obtained.

4.4.5.4 Problem Prioritization

Problems must be prioritized in the same way and for the same reasons as incidents – but the frequency and impact of related incidents must also be taken into account. The coding system described earlier in Table 4.1 (which combines impact with urgency to give an overall priority level) can be used to prioritize problems in the same way that it might be used for incidents, though the definitions and guidance to support staff on what constitutes a problem, and the related service targets at each level, must obviously be devised separately.

Problem prioritization should also take into account the severity of the problems. Severity in this context refers to how serious the problem is from an infrastructure perspective, for example:

- Can the system be recovered, or does it need to be replaced?
- How much will it cost?
- How many people, with what skills, will be needed to fix the problem?
- How long will it take to fix the problem?
- How extensive is the problem (e.g. how many CIs are affected)?

4.4.5.5 Problem Investigation and Diagnosis

An investigation should be conducted to try to diagnose the root cause of the problem – the speed and nature of this investigation will vary depending upon the impact, severity and urgency of the problem – but the appropriate level of resources and expertise should be applied to finding a resolution commensurate with the priority code allocated and the service target in place for that priority level.

There are a number of useful problem solving techniques that can be used to help diagnose and resolve problems – and these should be used as appropriate. Such techniques are described in more detail later in this section.

The CMS must be used to help determine the level of impact and to assist in pinpointing and diagnosing the exact point of failure. The Know Error Database (KEDB) should also be accessed and problem-matching techniques (such as key word searches) should be used to see if the problem has occurred before and, if so, to find the resolution.

It is often valuable to try to recreate the failure, so as to understand what has gone wrong, and then to try various ways of finding the most appropriate and cost-effective resolution to the problem. To do this effectively without causing further disruption to the users, a test system will be necessary that mirrors the production environment.

There are many problem analysis, diagnosis and solving techniques available and much research has been done in this area. Some of the most useful and frequently used techniques include:

- Chronological analysis**:** When dealing with a difficult problem, there are often conflicting reports about exactly what has happened and when. It is therefore very helpful briefly to document all events in chronological order – to provide a timeline of events. This often makes it possible to see which events may have been triggered by others – or to discount any claims that are not supported by the sequence of events.
- Pain Value Analysis**:** This is where a broader view is taken of the impact of an incident or problem, or incident/problem type. Instead of just analysing the number of incidents/problems of a particular type in a particular period, a more in-depth analysis is done to determine exactly what level of pain has been caused to the organization/business by these incidents/problems. A formula can be devised to calculate this pain level. Typically this might include taking into account:
    - The number of people affected
    - The duration of the downtime caused
    - The cost to the business (if this can be readily calculated or estimated).

  By taking all of these factors into account, a much more detailed picture of those incidents/problems or incident/problem types that are causing most pain can be determined – to allow a better focus on those things that really matter and deserve highest priority in resolving

- **Kepner and Tregoe:** Charles Kepner and Benjamin Tregoe developed a useful way of problem analysis which can be used formally to investigate deeper-rooted problems. They defined the following stages:
    - defining the problem
    - describing the problem in terms of identity, location, time and size
    - establishing possible causes
    - testing the most probable cause
    - verifying the true cause.

  The method is described in fuller detail in Appendix C.

- Brainstorming**:** It can often be valuable to gather together the relevant people, either physically or by electronic means, and to ‘brainstorm’ the

problem – with people throwing in ideas on what the potential cause may be and potential actions to resolve the problem. Brainstorming sessions can be very constructive and innovative but it is equally important that someone, perhaps the Problem Manager, documents the outcome and any agreed actions and keeps a degree of control in the session(s).

- Ishikawa Diagram**s:** Kaoru Ishikawa (1915–89), a leader in Japanese quality control, developed a method of documenting causes and effects which can be useful in helping identify where something may be going wrong, or be improved. Such a diagram is typically the outcome of a brainstorming session where problem solvers can offer suggestions. The main goal is represented by the trunk of the diagram, and primary factors are represented as branches. Secondary factors are then added as stems, and so on. Creating the diagram stimulates discussion and often leads to increased understanding of a complex problem. An example diagram is given in Appendix D.
- **Pareto Analysis:** This is a technique for separating important potential causes from more trivial issues. The following steps should be taken:

1. Form a table listing the causes and their frequency as a percentage.
2. Arrange the rows in the decreasing order of importance of the causes, i.e. the most important cause first.
3. Add a cumulative percentage column to the table. By this step, the chart should look something like Table 4.2, which illustrates 10 causes of network failure in an organization.

	Network failures		
Causes	Percentage of total	Computation	Cumulative %
Network Controller	35	0+35%	35
File corruption	26	35%+26%	61
Addressing conflicts	19	61%+19%	80
Server OS	6	80%+6%	86
Scripting error	5	86%+5%	91
Untested change	3	91%+3%	94
Operator error	2	94%+2%	96
Backup failure	2	96%+2%	98
Intrusion attempts	1	98%+1%	99
Disk failure	1	99%+1%	100

Table 4.2 Pareto cause ranking chart

4. Create a bar chart with the causes, in order of their percentage of total.
5. Superimpose a line chart of the cumulative percentages. The completed graph is illustrated in Figure 4.5.
6. Draw line at 80% on the y-axis parallel to the x-axis. Then drop the line at the point of intersection with the curve on the x-axis. This point on the x-axis separates the important causes and trivial causes. This line is represented as a dotted line in Figure 4.5.

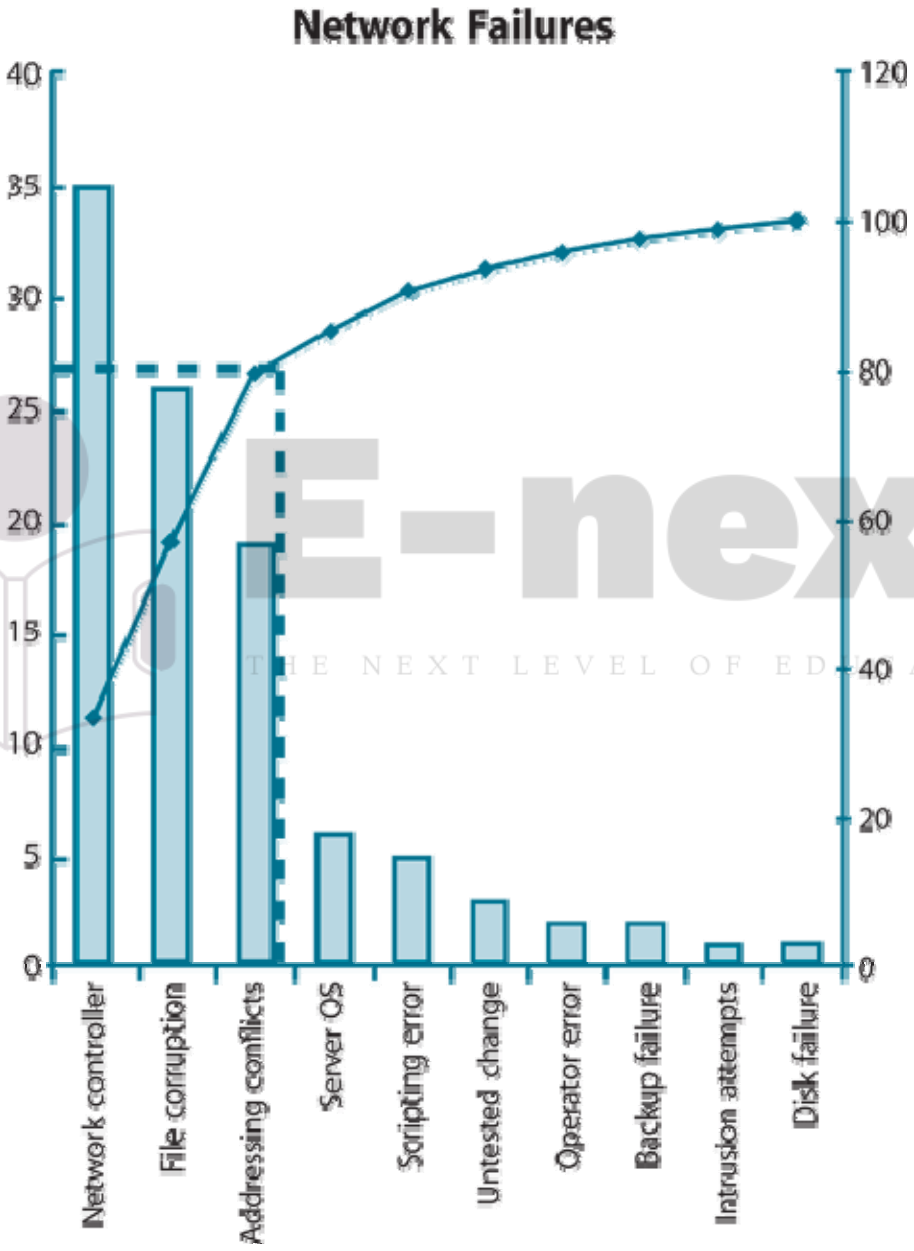

Figure 4.5 Important versus trivial causes

From this chart it is clear to see that there are three primary causes for network failure in the organization. These should therefore be targeted first.

4.4.5.6 Workarounds

In some cases it may be possible to find a workaround to the incidents caused by the problem – a temporary way of overcoming the difficulties. For example, a manual amendment may be made to an input file to allow a program to complete its run successfully and allow a billing process to complete satisfactorily, but it is important that work on a permanent resolution continues where this is justified – in this example the reason for the file becoming corrupted in the first place must be found and corrected to prevent this happening again.

In cases where a workaround is found, it is therefore important that the problem record remains open, and details of the workaround are always documented within the Problem Record.

4.4.5.7 Raising a Known Error Record

As soon as the diagnosis is complete, and particularly where a workaround has been found (even though it may not yet be a permanent resolution), a Known Error Record must be raised and placed in the Known Error Database – so that if further incidents or problems arise, they can be identified and the service restored more quickly.

However, in some cases it may be advantageous to raise a Known Error Record even earlier in the overall process – just for information purposes, for example – even though the diagnosis may not be complete or a workaround found, so it is inadvisable to set a concrete procedural point exactly when a Known Error Record must be raised. It should be done as soon as it becomes useful to do so!

The Known Error Database and the way it should be used are described in more detail in paragraph 4.4.7.2.

4.4.5.8 Problem resolution

Ideally, as soon as a solution has been found, it should be applied to resolve the problem – but in reality safeguards may be needed to ensure that this does not cause further difficulties. If any change in functionality is required this will require an RFC to be raised and approved before the resolution can be applied. If the problem is very serious and an urgent fix is needed for business reasons, then an Emergency RFC should be handled by the Emergency Change Advisory Board (ECAB). Otherwise, the RFC should follow the established Change Management process for that type of change – and the resolution should be applied only when the change has been approved and scheduled for release. In the meantime, the KEDB should be used to help resolve quickly any further occurrences of the incidents/problems that occur.

Note: There may be some problems for which a Business Case for resolution cannot be justified (e.g. where the impact is limited but the cost of resolution would be extremely high). In such cases a decision may be taken to leave the Problem Record open but to use a workaround description in the Known Error Record to detect and resolve any recurrences quickly. Care should be taken to use the appropriate code to flag the open Problem Record so that it does not count against the performance of the team performing the process and so that unauthorized rework does not take place.

4.4.5.9 Problem Closure

When any change has been completed (and successfully reviewed), and the resolution has been applied, the Problem Record should be formally closed – as should any related Incident Records that are still open. A check should be performed at this time to ensure that the record contains a full historical description of all events – and if not, the record should be updated.

The status of any related Known Error Record should be updated to shown that the resolution has been applied.

4.4.5.10 Major Problem Review

After every major problem (as determined by the organization's priority system), while memories are still fresh a review should be conducted to learn any lessons for the future. Specifically, the review should examine:

- Those things that were done correctly
- Those things that were done wrong
- What could be done better in the future
- How to prevent recurrence
- Whether there has been any third-party responsibility and whether follow-up actions are needed.

Such reviews can be used as part of training and awareness activities for support staff – and any lessons learned should be documented in appropriate procedures, work instructions, diagnostic scripts or Known Error Records. The Problem Manager facilitates the session and documents any agreed actions.

The knowledge learned from the review should be incorporated into a service review meeting with the business customer to ensure the customer is aware of the actions taken and the plans to prevent future major incidents from occurring. This helps to improve customer satisfaction and assure the business that Service Operations is handling major incidents responsibly and actively working to prevent their future recurrence.

4.4.5.11 Errors detected in the development environment

It is rare for any new applications, systems or software releases to be completely error-free. It is more likely that during testing of such new applications, systems or releases a prioritization system will be used to eradicate the more serious faults, but it is possible that minor faults are not rectified – often because of the balance that has to be made between delivering new functionality to the business as quickly as possible and ensuring totally fault-free code or components.

Where a decision is made to release something into the production environment that includes known deficiencies, these should be logged as Known Errors in the KEDB, together with details of workarounds or resolution activities. There should be a formal step in the testing sign-off that ensures that this handover always takes place (see Service Transition publication).

Experience has shown if this does not happen, it will lead to far higher support costs when the users start to experience the faults and raise incidents that have to be re-diagnosed and resolved all over again!

4.4.6 Triggers, input and output/inter-process interfaces

The vast majority of Problem Records will be triggered in reaction to one or more incidents, and many will be raised or initiated via Service Desk staff. Other Problem Records, and corresponding Known Error Records, may be triggered in testing, particularly the latter stages of testing such as User Acceptance Testing/Trials (UAT), if a decision is made to go ahead with a release even though some faults are known. Suppliers may trigger the need for some Problem Records through the notification of potential faults or known deficiencies in their products or services (e.g. a warning may be given regarding the use of a particular CI and a Problem Record may be raised to facilitate the investigation by technical staff of the condition of such CIs within the organization's IT Infrastructure).

The primary relationship between Incident and Problem Management has been discussed in detail in paragraphs 4.2.6 and 4.4.5.1. Other key interfaces include the following:

- Service Transition
  - Change Management**:** Problem Management ensures that all resolutions or workarounds that require a change to a CI are submitted through Change Management through an RFC. Change Management will monitor the progress of these changes and keep Problem Management advised. Problem Management is also involved in rectifying the situation caused by failed changes.
  - Configuration Management**:** Problem Management uses the CMS to identify faulty CIs and also to determine the impact of problems and resolutions. The CMS can also be used to form the basis for the KEDB and hold or integrate with the Problem Records.

    - Release and Deployment Management**:** Is responsible for rolling problem fixes out into the live environment. It also assists in ensuring that the associated known errors are transferred from the development Known Error Database into the live Known Error Database. Problem Management will assist in resolving problems caused by faults during the release process.
- Service Design
    - Availability Management**:** Is involved with determining how to reduce downtime and increase uptime. As such, it has a close relationship with Problem Management, especially the proactive areas. Much of the management information available in Problem Management will be communicated to Availability Management.
    - Capacity Management**:** Some problems will require investigation by Capacity Management teams and techniques, e.g. performance issues. Capacity Management will also assist in assessing proactive measures. Problem Management provides management information relative to the quality of decisions made during the Capacity Planning process.
    - IT service **Continuity:** Problem Management acts as an entry point into IT Service Continuity Management where a significant problem is not resolved before it starts to have a major impact on the business.
- Continual Service Improvement
    - Service Level Management**:** The occurrence of incidents and problems affects the level of service delivery measured by SLM. Problem Management contributes to improvements in service levels, and its management information is used as the basis of some of the SLA review components. SLM also provides parameters within which Problem Management works, such as impact information and the effect on services of proposed resolutions and proactive measures.
- Service Strategy
    - Financial Management**:** Assists in assessing the impact of proposed resolutions or workarounds, as well as Pain Value Analysis. Problem Management provides management information about the cost of resolving and preventing problems, which is used as input into the budgeting and accounting systems and Total Cost of Ownership calculations.

4.4.7 Information Management

4.4.7.1 CMS

The CMS will hold details of all of the components of the IT Infrastructure as well as the relationships between these components. It will act as a valuable source for problem diagnosis and for evaluating the impact of problems (e.g. if this disk

is down, what data is on that disk; which services use that data; which users use those services?). As it will also hold details of previous activities, it can also be used as a valuable source of historical data to help identify trends or potential weaknesses – a key part of proactive Problem Management (see Continual Service Improvement publication).

4.4.7.2 Known Error Database

The purpose of a Known Error Database is to allow storage of previous knowledge of incidents and problems – and how they were overcome – to allow quicker diagnosis and resolution if they recur.

The Known Error Record should hold exact details of the fault and the symptoms that occurred, together with precise details of any workaround or resolution action that can be taken to restore the service and/or resolve the problem. An incident count will also be useful to determine the frequency with which incidents are likely to recur and influence priorities, etc.

It should be noted that a Business Case for a permanent resolution for some problems may not exist. For example, if a problem does not cause serious disruption and a workaround exists and/or the cost of resolving the problem far outweighs the benefits of a permanent resolution – then a decision may be taken to tolerate the existence of the problem. However, it will still be desirable to diagnose and implement a workaround as quickly as possible, which is where the KEDB can be of assistance.

It is essential that any data put into the database can be quickly and accurately retrieved. The Problem Manager should be fully trained and familiar with the search methods/algorithms used by the selected database and should carefully ensure that when new records are added, the relevant search key criteria are correctly included.

Care should be taken to avoid duplication of records (i.e. the same problem described in two or more ways as separate records). To avoid this, the Problem Manager should be the only person able to enter a new record. Other support groups should be allowed, indeed encouraged, to propose new records, but these should be vetted by the Problem Manager before entry to the KEDB. In large organizations where Problem Management staff exist in multiple locations but a single KEDB is used (recommended!), a procedure must be agreed between all Problem Management staff to ensure that such duplication cannot occur. This may involve designating just one staff member as the central KEDB Manager.

The KEDB should be used during the Incident and Problem Diagnosis phases to try to speed up the resolution process – and new records should be added as quickly as possible when a new problem has been identified and diagnosed.

All support staff should be fully trained and conversant with the value that the KEDB can offer and the way it should be used. They should be able readily to retrieve and use data.

Note: Some tools/implementations may choose to delineate Known Errors simply by changing a field in the original Problem Record. This is acceptable provided the same level of functionality is available.

The KEDB, like the CMS, forms part of a larger Service Knowledge Management System (SKMS) illustrated in Figure 4.6. More information on the SKMS can be found in the Service Transition publication.

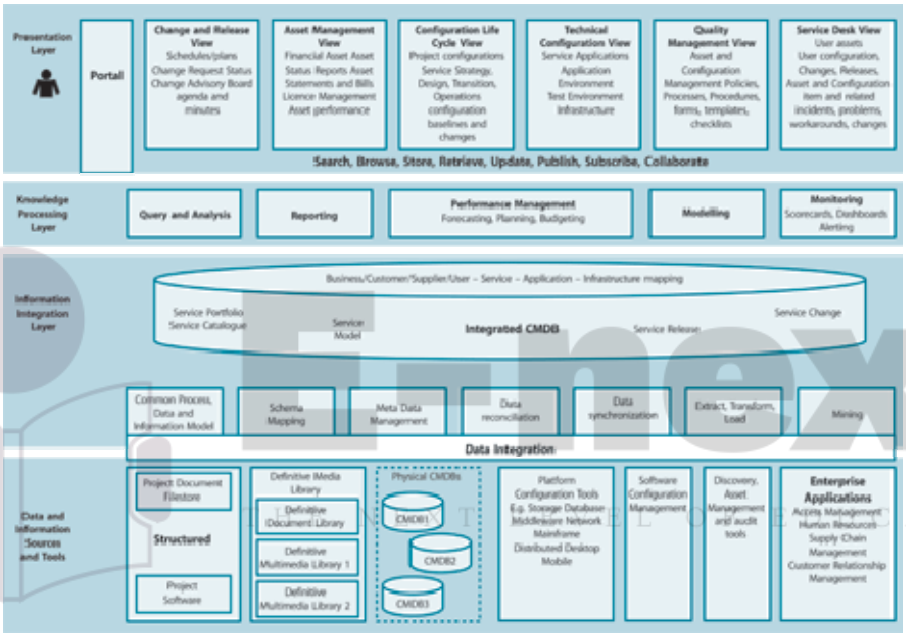

Figure 4.6 Service Knowledge Management System

4.4.8 Metrics

The following metrics should be used to judge the effectiveness and efficiency of the Problem Management process, or its operation:

- The total number of problems recorded in the period (as a control measure)
- The percentage of problems resolved within SLA targets (and the percentage that are not!)
- The number and percentage of problems that exceeded their target resolution times
- The backlog of outstanding problems and the trend (static, reducing or increasing?)
- The average cost of handling a problem

- The number of major problems (opened and closed and backlog)
- The percentage of Major Problem Reviews successfully performed
- The number of Known Errors added to the KEDB
- The percentage accuracy of the KEDB (from audits of the database)
- The percentage of Major Problem Reviews completed successfully and on time.

All metrics should be broken down by category, impact, severity, urgency and priority level and compared with previous periods.

4.4.9 Challenges, Critical Success Factors and risks

A major dependency for Problem Management is the establishment of an effective Incident Management process and tools. This will ensure that problems are identified as soon as possible and that as much work is done on pre-qualification as possible. However, it is also critical that the two processes have formal interfaces and common working practices. This implies the following:

- Linking Incident and Problem Management tools
- The ability to relate Incident and Problem Records
- The second- and third-line staff should have a good working relationship with staff on the first line
- Making sure that business impact is well understood by all staff working on problem resolution.

In addition it is important that Problem Management is able to use all Knowledge and Configuration Management resources available.

Another CSF is the ongoing training of technical staff in both technical aspects of their job as well as the business implications of the services they support and the processes they use

4.5 Access Management

Access Management is the process of granting authorized users the right to use a service, while preventing access to non-authorized users. It has also been referred to as Rights Management or Identity Management in different organizations.

4.5.1 Purpose/goal/objective

Access Management provides the right for users to be able to use a service or group of services. It is therefore the execution of policies and actions defined in Security and Availability Management.

4.5.2 Scope

Access Management is effectively the execution of both Availability and Information Security Management, in that it enables the organization to manage the confidentiality, availability and integrity of the organization's data and intellectual property.

Access Management ensures that users are given the right to use a service, but it does not ensure that this access is available at all agreed times – this is provided by Availability Management.

Access Management is a process that is executed by all Technical and Application Management functions and is usually not a separate function. However, there is likely to be a single control point of coordination, usually in IT Operations Management or on the Service Desk.

Access Management can be initiated by a Service Request through the Service Desk.

4.5.3 Value to business

Access Management provides the following value:

- Controlled access to services ensures that the organization is able to maintain more effectively the confidentiality of its information
- Employees have the right level of access to execute their jobs effectively
- There is less likelihood of errors being made in data entry or in the use of a critical service by an unskilled user (e.g. production control systems)
- The ability to audit use of services and to trace the abuse of services
- The ability more easily to revoke access rights when needed – an important security consideration
- May be needed for regulatory compliance (e.g. SOX, HIPAA, COBIT).

4.5.4 Policies/principles/basic concepts

Access Management is the process that enables users to use the services that are documented in the Service Catalogue. It comprises the following basic concepts:

- **Access** refers to the level and extent of a service's functionality or data that a user is entitled to use.
- Identity refers to the information about them that distinguishes them as an individual and which verifies their status within the organization. By definition, the Identity of a user is unique to that user. (This is covered in more detail in paragraph 4.5.7.1.)
- **Rights** (also called privileges) refer to the actual settings whereby a user is provided access to a service or group of services. Typical rights, or levels of access, include read, write, execute, change, delete.
- **Services or service groups**. Most users do not use only one service, and users performing a similar set of activities will use a similar set of services. Instead of providing access to each service for each user separately, it is more efficient to be able to grant each user – or group of users – access to the whole set of services that they are entitled to use at the same time. (This is discussed in more detail in paragraph 4.5.7.2.)
- **Directory Services** refers to a specific type of tool that is used to manage access and rights. These are discussed in section 5.8.

4.5.5 Process activities, methods and techniques

4.5.5.1 Requesting access

Access (or restriction) can be requested using one of any number of mechanisms, including:

- A standard request generated by the Human Resource system. This is generally done whenever a person is hired, promoted, transferred or when they leave the company
- A Request for Change
- A Service Request submitted via the Request Fulfilment system
- By executing a pre-authorized script or option (e.g. downloading an application from a staging server as and when it is needed).

Rules for requesting access are normally documented as part of the Service Catalogue.

4.5.5.2 Verification

Access Management needs to verify every request for access to an IT service from two perspectives:

- That the user requesting access is who they say they are
- That they have a legitimate requirement for that service.

The first category is usually achieved by the user providing their username and password. Depending on the organization's security policies, the use of the username and password are usually accepted as proof that the person is a legitimate user. However, for more sensitive services further identification may be required (biometric, use of an electronic access key or encryption device, etc.).

The second category will require some independent verification, other than the user's request. For example:

- Notification from Human Resources that the person is a new employee and requires both a username and access to a standard set of services
- Notification from Human Resources that the user has been promoted and requires access to additional resources
- Authorization from an appropriate (defined in the process) manager
- Submission of a Service Request (with supporting evidence) through the Service Desk
- Submission of an RFC (with supporting evidence) through Change Management, or execution of a pre-defined Standard Change
- A policy stating that the user may have access to an optional service if they need it.

For new services the Change Record should specify which users or groups of users will have access to the Service. Access Management will then check to see that all the users are still valid and automatically provide access as specified in the RFC.

4.5.5.3 Providing rights

Access Management does not decide who has access to which IT services. Rather, Access Management executes the policies and regulations defined during Service Strategy and Service Design. Access Management enforces decisions to restrict or provide access, rather than making the decision.

As soon as a user has been verified, Access Management will provide that user with rights to use the requested service. In most cases this will result in a request to every team or department involved in supporting that service to take the necessary action. If possible, these tasks should be automated.

The more roles and groups that exist, the more likely that Role Conflict will arise. Role Conflict in this context refers to a situation where two specific roles or groups, if assigned to a single user, will create issues with separation of duties or conflict of interest. Examples of this include:

- One role requires detailed access, while another role prevents that access
- Two roles allow a user to perform two tasks that should not be combined (e.g. a contractor can log their time sheet for a project and then approve all payment on work for the same project).

Role Conflict can be avoided by careful creation of roles and groups, but more often they are caused by policies and decisions made outside of Service Operation – either by the business or by different project teams working during Service Design. In each case the conflict must be documented and escalated to the stakeholders to resolve.

Whenever roles and groups are defined, it is possible that they could be defined too broadly or too narrowly. There will always be users who need something slightly different from the pre-defined roles. In these cases, it is possible to use standard roles and then add or subtract specific rights as required – similar to the concept of Baselines and Variants in Configuration Management (see Service Transition publication). However, the decision to do this is not in the hands of individual operational staff members. Each exception should be coordinated by Access Management and approved through the originating process.

Access Management should perform a regular review of the roles and groups that it has created and manage to ensure that they are appropriate for the services that IT delivers and supports – and obsolete or unwanted roles/groups should be removed.

4.5.5.4 Monitoring identity status

As users work in the organization, their roles change and so also do their needs to access services. Examples of changes include:

- **Job changes.** In this case the user will possibly need access to different or additional services.
- **Promotions or demotions.** The user will probably use the same set of services, but will need access to different levels of functionality or data.
- **Transfers.** In this situation, the user may need access to exactly the same set of services, but in a different region with different working practices and different sets of data.
- **Resignation or death.** Access needs to be completely removed to prevent the username being used as a security loophole.
- **Retirement.** In many organizations, an employee who retires may still have access to a limited set of services, including benefits systems or systems that allow them to purchase company products at a reduced rate.
- **Disciplinary action.** In some cases the organization will require a temporary restriction to prevent the user from accessing some or all of the services that they would normally have access to. There should be a

feature in the process and tools to do this, rather than having to delete and reinstate the user's access rights.

- **Dismissals.** Where an employee or contractor is dismissed, or where legal action is taken against a customer (for example for defaulting on payment for products purchased on the Internet), access should be revoked immediately. In addition, Access Management, working together with Information Security Management, should take active measures to prevent and detect malicious action against the organization from that user.

Access Management should understand and document the typical User Lifecycle for each type of user and use it to automate the process. Access Management tools should provide features that enable a user to be moved from one state to another, or from one group to another, easily and with an audit trail.

4.5.5.5 Logging and tracking access

Access Management should not only respond to requests. It is also responsible for ensuring that the rights that they have provided are being properly used.

In this respect, Access Monitoring and Control must be included in the monitoring activities of all Technical and Application Management functions and all Service Operation processes.

Exceptions should be handled by Incident Management, possibly using Incident Models specifically designed to deal with abuse of access rights. It should be noted that the visibility of such actions should be restricted. Making this information available to all who have access to the Incident Management system will expose vulnerabilities.

Information Security Management plays a vital role in detecting unauthorized access and comparing it with the rights that were provided by Access Management. This will require Access Management involvement in defining the parameters for use in Intrusion Detection tools.

Access Management may also be required to provide a record of access for specific Services during forensic investigations. If a user is suspected of breaches of policy, inappropriate use of resources, or fraudulent use of data, Access Management may be required to provide evidence of dates, times and even content of that user's access to specific Services. This is normally provided by the Operational staff of that service, but working as part of the Access Management process.

4.5.5.6 Removing or restricting rights

Just as Access Management provides rights to use a Service, it is also responsible for revoking those rights. Again, this is not a decision that it makes on its own. Rather, it will execute the decisions and policies made during Service Strategy and Design and also decisions made by managers in the organization.

Removing access is usually done in the following circumstances:

- Death
- Resignation
- Dismissal
- When the user has changed roles and no longer requires access to the service
- Transfer or travel to an area where different regional access applies.

In other cases it is not necessary to remove access, but just to provide tighter restrictions. These could include reducing the level, time or duration of access. Situations in which access should be restricted include:

- When the user has changed roles or been demoted and no longer requires the same level of access
- When the user is under investigation, but still requires access to basic services, such as e-mail. In this case their e-mail may be subject to additional scanning (but this would need to be handled very carefully and in full accordance with the organization's security policy)
- When a user is away from the organization on temporary assignment and will not require access to that service for some time.

4.5.6 Triggers, input and output/inter-process interfaces

Access Management is triggered by a request for a user or users to access a service or group of services. This could originate from any of the following:

- **An RFC.** This is most frequently used for large-scale service introductions or upgrades where the rights of a significant number of users need to be updated as part of the project.
- **A** Service Request**.** This is usually initiated through the Service Desk, or directly into the Request Fulfilment system, and executed by the relevant Technical or Application Management teams.
- A request from the appropriate **Human Resources Management** personnel (which should be channelled via the Service Desk). This is usually generated as part of the process for hiring, promoting, relocating and termination or retirement.
- A request from the **manager of a department**, who could be performing an HR role, or who could have made a decision to start using a service for the first time.

Access Management should be linked to the Human Resource processes to verify the user's identify as well as to ensure that they are entitled to the services being requested.

Information Security Management is a key driver for Access Management as it will provide the security and data protection policies and tools needed to execute Access Management.

Change Management plays an important role as the means to control the actual requests for access. This is because any request for access to a service is a change, although it is usually processed as a Standard Change or Service Request (possibly using a model) once the criteria for access have been agreed through SLM.

SLM maintains the agreements for access to each service. This will include the criteria for who is entitled to access each service, what the cost of that access will be, if appropriate and what level of access will be granted to different types of user (e.g. managers or staff).

There is also a strong relationship between Access Management and Configuration Management. The CMS can be used for data storage and interrogated to determine current access details.

4.5.7 Information Management

4.5.7.1 Identity

The identity of a user is the information about them that distinguishes them as an individual and which verifies their status within the organization. By definition, the identity of a user is unique to that user. Since there are cases where two users share a common piece of information (e.g. they have the same name), identity is usually established using more than one piece of information, for example:

- Name
- Address
- Contact details, e.g. telephone, e-mail address, etc.
- Physical documentation, e.g. driver's licence, passport, marriage certificate, etc.
- Numbers that refer to a document or an entry in a database, e.g. employee number, tax number, government identity number, driver's licence number, etc.
- Biometric information, e.g. fingerprints, retinal images, voice recognition patterns, DNA, etc.
- Expiration date (if relevant).

A user identity is provided to anyone with a legitimate requirement to access IT services or organizational information. These could include:

- Employees
- Contractors
- Vendor staff (e.g. account managers, support personnel, etc.)
- Customers (especially when purchasing products or services over the Internet).

Most organizations will verify a user's identity before they join the organization by requesting a subset of the above information. The more secure the organization, the more types of information are required and the more thoroughly they are checked.

Many organizations will be faced with the need to provide access rights to temporary or occasional staff or contractors/suppliers. The management of access to such personnel often proves problematic – closing access after use is often as difficult to manage, or more so, than providing access initially. Well-defined procedures between IT and HR should be established that include fail-safe checks that ensure access rights are removed immediately they are no longer justified or required.

When a user is granted access to an application, it should already have been established by the organization (usually the Human Resources or Security Department) that the user is who they say they are.

At this point, all that information is filed and the file is associated with a corporate identity, usually an employee or contractor number and an identity that can be used to access corporate resources and information, usually a user identity or 'username' and an associated password.

4.5.7.2 Users, groups, roles and service groups

While each user has an individual identity, and each IT service can be seen as an entity in its own right, it is often helpful to group them together so that they can be managed more easily. Sometimes the terms 'user profile' or 'user template' or 'user role' are used to describe this type of grouping.

Most organizations have a standard set of services for all individual users, regardless of their position or job (excluding customers – who do not have any visibility to internal services and processes). These will include services such as messaging, office automation, Desktop Support, telephony, etc. New users are automatically provided with rights to use these services.

However, most users also have some specialized role that they perform. For example, in addition to the standard services, the user also performs a Marketing

Management role, which requires that they have access to some specialized marketing and financial modelling tools and data.

Some groups may have unique requirements – such as field or home workers who may have to dial in or use Virtual Private Network (VPN) connections, with security implications that may have to be more tightly managed.

To make it easier for Access Management to provide the appropriate rights, it uses a catalogue of all the roles in the organization and which services support each role. This catalogue of roles should be compiled and maintained by Access Management in conjunction with HR and will often be automated in the Directory Services tools (see section 5.8).

In addition to playing different roles, users may also belong to different groups. For example, all contractors are required to log their timesheets in a dedicated Time Card System, which is not used by employees. Access Management will assess all the roles that a user plays as well as the groups that they belong to and ensure that they provide rights to use all associated services.

Note: All data held on users will be subject to data protection legislation (this exists in most geographic locations in some form or other) so should be handled and protected as part of the organization's security procedures.

4.5.8 Metrics

Metrics that can be used to measure the efficiency and effectiveness of Access Management include:

- Number of requests for access (Service Request, RFC, etc.)
- Instances of access granted, by service, user, department, etc.
- Instances of access granted by department or individual granting rights
- Number of incidents requiring a reset of access rights
- Number of incidents caused by incorrect access settings.

4.5.9 Challenges, Critical Success Factors and risks

Conditions for successful Access Management include:

- The ability to verify the identity of a user (that the person is who they say they are)
- The ability to verify the identity of the approving person or body
- The ability to verify that a user qualifies for access to a specific service
- The ability to link multiple access rights to an individual user
- The ability to determine the status of the user at any time (e.g. to determine whether they are still employees of the organization when they log on to a system)

- The ability to manage changes to a user's access requirements
- The ability to restrict access rights to unauthorized users
- A database of all users and the rights that they have been granted.

4.6 Operational activities of processes covered in other lifecycle phases

4.6.1 Change Management

Change Management is primarily covered in the Service Transition publication, but there are some aspects of Change Management which Service Operation staff will be involved with on a day-to-day basis. These include:

- Raising and submitting RFCs as needed to address Service Operation issues
- Participating in CAB or CAB/EC meetings to ensure that Service Operation risks, issues and views are taken into account
- Implementing changes as directed by Change Management where they involve Service Operation component or services
- Backing out changes as directed by Change Management where they involve Service Operation component or services
- Helping define and maintain change models relating to Service Operation components or services
- Receiving change schedules and ensuring that all Service Operation staff are made aware of and prepared for all relevant changes
- Using the Change Management process for standard, operational-type changes.

4.6.2 Configuration Management

Configuration Management is primarily covered in the Service Transition publication, but there are some aspects of Configuration Management which Service Operation staff will be involved with on a day-to-day basis. These include:

- Informing Configuration Management of any discrepancies found between any CIs and the CMS
- Making any amendments necessary to correct any discrepancies, under the authority of Configuration Management, where they involve any Service Operation components or services.

Responsibility for updating the CMS remains with Configuration Management, but in some cases Operations staff might be asked, under the direction of Configuration Management, to update relationships, or even to add new CIs or mark CIs as ‘disposed’ in the CMS, if these updates are related to operational activities actually performed by Operations staff.

4.6.3 Release and Deployment Management

Release and Deployment Management is primarily covered in the Service Transition publication, but there are some aspects of this process which Service Operation staff will be involved with on a day-to-day basis. These may include:

- Actual implementation actions regarding the deployment of new releases, under the direction of Release and Deployment Management, where they relate to Service Operation components or services
- Participation in the planning stages of major new releases to advise on Service Operation issues
- The physical handling of CIs from/to the DML as required to fulfil their operational roles – while adhering to relevant Release and Deployment Management procedures, such as ensure that all items are properly booked out and back in.

4.6.4 Capacity Management

Capacity Management should operate at three levels: Business Capacity Management, Service Capacity Management and Resource Capacity Management.

- **Business Capacity Management** involves working with the business to plan and anticipate both longer-term strategic issues and shorter-term tactical initiatives that are likely to have an impact on IT capacity.
- **Service Capacity Management** is about understanding the characteristics of each of the IT services, and then the demands that different types of users or transactions have on the underlying infrastructure – and how these vary over time and might be impacted by business change.
- Resource **Capacity Management** involves understanding the performance characteristics and capabilities and current utilization levels of all the technical components (CIs) that make up the IT Infrastructure, and predicting the impact of any changes or trends.

Many of these activities are of a strategic or longer-term planning nature and are covered in the Service Strategy, Service Design and Service Transition publications. However, there are a number of operational Capacity Management activities that must be performed on a regular ongoing basis as part of Service Operation. These include the following.

4.6.4.1 Capacity and Performance Monitoring

All components of the IT Infrastructure should be continually monitored (in conjunction with Event Management) so that any potential problems or trends can be identified before failures or performance degradation occurs. Ideally, such

monitoring should be automated and thresholds should be set so that exception alerts are raised in good time to allow appropriate avoiding or recovery action to be taken before adverse impact occurs.

The components and elements to be monitored will vary depending upon the infrastructure in use, but will typically include:

- CPU utilization (overall and broken down by system/service usage)
- Memory utilization
- IO rates (physical and buffer) and device utilization
- Queue length (maximum and average)
- File store utilization (disks, partitions, segments)
- Applications (throughput rates, failure rates)
- Databases (utilization, record locks, indexing, contention)
- Network transaction rates, error and retry rates
- Transaction response time
- Batch duration profiles
- Internet/intranet site/page hit rates
- Internet response times (external and internal to firewalls)
- Number of system/application log-ons and concurrent users
- Number of network nodes in use, and utilization levels.

There are different kinds of monitoring tools needed to collect and interpret data at each level. For example, some tools will allow performance of business transactions to be monitored, while others will monitor CI behaviour.

Capacity Management must set up and calibrate alarm thresholds (where necessary in conjunction with Event Management, as it is often Event Monitoring tools that may be used) so that the correct alert levels are set and that any filtering is established as necessary so that only meaningful events are raised. Without such filtering it is possible that ‘information only’ alerts can obscure more significant alerts that require immediate attention. In addition, it is possible for serious failures to cause ‘alert storms’ due to very high volumes of repeat alerts, which again must be filtered so that the most meaningful messages are not obscured.

It may be appropriate to use external, third-party, monitoring capabilities for some CIs or components of the IT Infrastructure (e.g. key internet sites/pages). Capacity Management should be involved in helping specify and select any such monitoring capabilities and in integrating the results or any alerts with other monitoring and handling systems.

Capacity Management must work with all appropriate support groups to make decisions on where alarms are routed and on escalation paths and timescales. Alerts should be logged to the Service Desk as well as to appropriate support staff, so that appropriate Incident Records can be raised so a permanent record

of the event exists – and Service Desk staff have a view of how well the support group(s) are dealing with the fault and can intervene if necessary.

Manufacturers' claimed performance capabilities and agreed service level targets, together with actual historical monitored performance and capacity data, should be used to set alert levels. This may need to be an iterative process initially, performing some trial-and-error adjustments until the correct levels are achieved.

Note: Capacity Management may have to become involved in the capacity requirements and capabilities of IT Service Management. Whether the organization has enough Service Desk staff to handle the rate of incidents; whether the CAB structure can handle the number of changes it is being asked to review and approve; whether support tools can handle the volume of data being gathered are Capacity Management issues, which the Capacity Management team may be asked to help investigate and answer.

4.6.4.2 Handling capacity- or performance-related incidents

If an alert is triggered, or an incident is raised at the Service Desk, caused by a current or ongoing Capacity or Performance Management problem, Capacity Management must become involved to identify the cause and find a resolution. Working together with appropriate technical support groups, and alongside Problem Management, all necessary investigations must be performed to detect exactly what has gone wrong and what is needed to correct the situation.

It may be necessary to switch to more detailed monitoring during the investigation phase to determine the exact cause. Monitoring is often set at a 'background' level during normal circumstances due to the large amount of data that can be generated and to avoid placing too high a burden on the IT Infrastructure – but when specific difficulties are being investigated more detailed monitoring may be needed to pinpoint the exact cause.

When a solution, or potential solution, has been found, any changes necessary to resolve the problem must be approved via formal Change Management prior to implementation. If the fault is causing serious disruption and an urgent resolution is needed, the urgent change process should be used. It is very important that no 'tuning' takes place without submission through Change Management, as even apparently small adjustments can often have very large cumulative effects – sometimes across the entire IT Infrastructure.

4.6.4.3 Capacity and performance trends

Capacity Management has a role to play in identifying any capacity or performance trends as they become discernible. Further details of actions

needed to address such trends are included in the Continual Service Improvement publication.

4.6.4.4 Storage of Capacity Management data

Large amounts of data are usually generated through capacity and performance monitoring. Monitoring of meters and tables of just a few Kbytes each can quickly grown into huge files if many components are being monitored at relatively short intervals. Another problem with very short-term monitoring is that it is not possible to gather meaningful information without looking over a longer period. For example, a single snapshot of a CPU will show the device to be either 'busy' or 'idle' – but a summary over, say, a 5-minute period will show the average utilization level over that period, which is a much more meaningful measure of whether the device is able to work comfortably, or whether potential performance problems are likely to occur.

In any organization it is likely that the monitoring tools used will vary greatly – with a combination of system-specific tools, many of them part of the basic operating system, and specialist monitoring tools being used. In order to coordinate the data being generated and allow the retention of meaningful data for analysis and trending purposes, some form of central repository for holding this summary data is needed: a Capacity Management Information System (CMIS).

The format, location and design of such a database should be planned and implemented in advance – see the Service Design publication for further details – but there will be some operational aspects to handle, such as database housekeeping and backups.

4.6.4.5 Demand Management

Demand Management is the name given to a number of techniques that can be used to modify demand for a particular resource or service. Some techniques for Demand Management can be planned in advance– and these are covered in more detail in the Service Design publication. However, there are other aspects of Demand Management that are of a more operational nature, requiring shorter-term action.

If, for example, the performance of a particular service is causing concern, and short-term restrictions on concurrency of users are needed to allow performance improvements for a smaller restricted group, then Service Operation functions will have to take action to implement such restrictions – usually accompanied by concurrent action to implement the logging-out of users who have been inactive for an agreed period of time to free up resources for others.

4.6.4.6 Workload Management

There may be occasions when optimization of infrastructure resources is needed to maintain or improve performance or throughput. This can often be done through Workload Management, which is a generic term to cover such actions as:

- Rescheduling a particular service or workload to run at a different time of day, or day of the week etc. (usually away from peak-times to off-peak windows) – which will often mean having to make adjustments to job-scheduling software.
- Moving a service or workload from one location or set of CIs to another – often to balance utilization or traffic.
- Technical Virtualization: setting up and using virtualization systems to allow movement of processing around the infrastructure to give better performance/resilience in a dynamic fashion.
- Limiting or moving demand for resources through Demand Management techniques (see above and also the Service Design publication).

It will only be possible to manage workloads effectively if a good understanding exists of which workloads will run at what time and how much resource utilization each workload places upon the IT Infrastructure. Diligent monitoring and analysis of workloads is therefore needed on an ongoing operational basis.

4.6.4.7 Modelling and applications sizing

Modelling and/or sizing of new services and/or applications must, where appropriate, be done during the planning and transition phases – see the Service Design and Service Transition publications. However, the Service Operation functions have a role to play in evaluating the accuracy of the predictions and feeding back any issues or discrepancies.

4.6.4.8 Capacity Planning

During Service Design and Service Transition, the capacity requirements of IT services are calculated. A forward-looking capacity plan should be maintained and regularly updated and Service Operation will have a role to play in this. Such a plan should look forward up to two years or more, but should be reviewed regularly every three to 12 months, depending upon volatility and resources available.

The plan should be linked to the organization's financial planning cycle, so that any required expenditure for infrastructure upgrades, enhancements or additions can be included in budget estimates and approved in advance.

The plan should predict the future but must also examine and report upon previous predictions, particularly to give some confidence in further predictions.

Where any discrepancies have been encountered, these should be explained and future remedial action described.

The Capacity Plan might typically cover:

- Current performance and utilization details, with recent trends for all key CIs, including
  - Backbone networks
  - LANs
  - Mainframes (if still used)
  - Key servers
  - Main data storage devices
  - Selected (representative) desktop and laptop equipment
  - Key websites
  - Key databases
  - Key applications
  - Operational capacity – electricity, floor space, environmental capacity (air condition), floor weighting, heat generation and output, electrical and water demand and supply etc.
  - Magnetic media.
- Estimated performance and utilization for all such CIs during the planning period (e.g. the next three months)
- Comparative data with previous estimates – to allow confidence in future estimates to be judged
- Reports on any specific capacity difficulties encountered in the past period, with details of recovery and preventive actions taken for the future
- Details of any required upgrades or procurements needed and planned for the future, with indicative costs and timescales.
- Any potential capacity risks that are likely – with suggested countermeasures should they arise.

4.6.5 Availability Management

During Service Design and Service Transition, IT services are designed for availability and recovery. Service Operation is responsible for actually making the IT service available to the specified users at the required time and at the agreed levels.

During Service Operation the IT teams and users are in the best position to detect whether services actually meet the agreed requirements and whether the design of these services is effective.

What seems like a good idea during the Design phase may not actually be practical or optimal. The experience of the users and operational functions makes them a primary input into the ongoing improvement of existing services and the design.

However, there are a number of challenges with gaining access to this knowledge:

- Most of the experiences of the operational teams and users are either informal, or spread across multiple sources.
- The process for collecting and collating this data needs to be formalized.
- Users and operational staff are usually fully occupied with their regular activities and tasks and it is very difficult for them to be involved in regular planning and design activities. One argument often made here is that if design is improved, the operational teams will be less busy resolving problems and will therefore have more time to be involved in design activities. However, practice shows that as soon as staff are freed up, they often become the target of workforce reduction exercises.

Having said this, there are three key opportunities for operational staff to be involved in Availability Improvement, since these are generally viewed as part of their ongoing responsibility:

- Review **of maintenance activities.** Service Design will define detailed maintenance schedules and activities, which are required to keep IT services functioning at the required level of performance and availability. Regular comparison of actual maintenance activities and times with the plans will highlight potential areas for improvement. One of the sources of this information is a review of whether Service Maintenance Objectives were met and, if not, why not.
- **Major problem reviews.** Problems could be the result of any number of factors, one of which is poor design. Problem reviews therefore may include opportunities to identify improvements to the design of IT services, which will include availability and capacity improvement.
- Involvement in **specific initiatives** using techniques such as Service Failure Analysis (SFA), Component Failure Impact Analysis (CFIA), or Fault Tree Analysis (FTA) or as members of Technical Observation (TO) activities – either as part of the follow-up to major problems or as part of an ongoing service improvement programme, in collaboration with dedicated Availability Management staff. These Availability Management techniques are explained in more detail in the Service Design publication.

There may be occasions when Operational Staff themselves need downtime of one or more services to enable them to conduct their operational or maintenance activities – which may impact on availability if not properly scheduled and managed. In such cases they must liaise with SLM and Availability Management staff – who will negotiate with the business/users, often using the Service Desk to perform this role, to agree and schedule such activities.

4.6.6 Knowledge Management

It is vitally important that all data and information that can be useful for future Service Operation activities are properly gathered, stored and assessed. Relevant data, metrics and information should be passed up on the management chain and to other Service Lifecycle phases so that it can feed into the knowledge and wisdom layers of the organization's Service Knowledge Management System, the structures of which have to be defined in Service Strategy and Service Design and refined in Continual Service Improvement (see other ITIL publications in this series).

Key repositories of Service Operation, which have been frequently mentioned elsewhere, are the CMS and the KEDB, but this must be widened out to include all of the Service Operation teams' and departments' documentation, such as operations manuals, procedures manuals, work instructions, etc.

4.6.7 Financial Management for IT services

Service Operation staff must participate in and support the overall IT budgeting and accounting system – and may be actively involved in any charging system that may be in place.

Proper planning is necessary so that capital expenditure (Capex) and operational expenditure (Opex) budget estimates can be prepared and agreed in good time to meet the budgetary cycles.

The Service Operation Manager must also be involved in regular, at least monthly, reviews of expenditure against budgets – as part of the ongoing IT budgeting and accounting process. Any discrepancies must be identified and necessary adjustments made. All committed expenditure must go through the organization's purchase order system so that commitments can be accrued and proper checks must be made on all goods received so that invoices and payments can be correctly authorized – or discrepancies investigated and rectified.

It should be noted that some proposed cost reductions by the business may actually increase IT costs, or at least unit costs. Care should therefore be taken to ensure that IT is involved in discussing all cost-saving measures and contribute to overall decisions. Financial Management is covered in detail in the Service Strategy publication.

4.6.8 IT Service Continuity Management

Service Operation functions are responsible for the testing and execution of system and service recovery plans as determined in the IT Service Continuity

Plans for the organization. In addition, managers of all Service Operation functions must be on the Business Continuity Central Coordination team.

This is discussed in detail in Service Strategy and Service Design and will not be repeated here, except to indicate that it is important that Service Operation functions must be involved in the following areas:

- Risk assessment, using its knowledge of the infrastructure and techniques such as CFIA and access to information in the CMS to identify single points of failure or other high-risk situations
- Execution of any Risk Management measures that are agreed, e.g. implementation of countermeasures, or increased resilience to components of the infrastructures, etc.
- Assistance in writing the actual recovery plans for systems and services under its control
- Participation in testing of the plans (such as involvement in off-site testing, simulations etc) on an ongoing basis under the direction of the IT Service Continuity Manager (ITSCM)
- Ongoing maintenance of the plans under the control of ITSCM and Change Management
- Participation in training and awareness campaigns to ensure that they are able to execute the plans and understand their roles in a disaster
- The Service Desk will play a key role in communicating with staff, customers and users during an actual disaster