Chapter 4:

Information Systems Operations and Business Resilience

Overview

Domain 4 Exam Content Outline
Learning Objectives/Task Statements
Suggested Resources for Further Study
Self-assessment Questions
Answers to Self-assessment Questions

Part A: Information Systems Operations

4.0 Introduction
4.1 Common Technology Components
4.2 IT Asset Management
4.3 Job Scheduling and Production Process Automation
4.4 System Interfaces
4.5 End-user Computing
4.6 Data Governance
4.7 Systems Performance Management
4.8 Problem and Incident Management
4.9 Change, Configuration, Release and Patch Management
4.10 IT Service Level Management.
4.11 Database Management

Part B: Business Resilience

4.12 Business Impact Analysis
4.13 System Resiliency
4.14 Data Backup, Storage and Restoration
4.15 Business Continuity Plan
4.16 Disaster Recovery Plans

Case Study

Case Study
Answers to Case Study Questions

OVERVIEW

Information systems operations and business resilience are important to provide assurance to users and management that the expected level of service will be delivered. Service level expectations are derived from the organization's business objectives. Information technology (IT) service delivery includes information systems (IS) operations, IT services and management of IS and the groups responsible for supporting them. Disruptions are also an often-unavoidable factor of doing business. Preparation is key to being able to continue business operations while protecting people, assets and reputation. Employing business resiliency tactics helps organizations address these issues and limit the impact.

This domain represents 23 percent of the CISA examination (approximately 34 questions).

DOMAIN 4 EXAM CONTENT OUTLINE

Part A: Information Systems Operations

1. Common Technology Components
2. IT Asset Management
3. Job Scheduling and Production Process Automation
4. System Interfaces
5. End-User Computing
6. Data Governance
7. Systems Performance Management
8. Problem and Incident Management
9. Change, Configuration, Release, and Patch Management
10. IT Service Level Management
11. Database Management

Part B. Business Resilience

1. Business Impact Analysis (BIA)

2. System Resiliency
3. Data Backup, Storage, and Restoration
4. Business Continuity Plan (BCP)
5. Disaster Recovery Plans (DRPs)

LEARNING OBJECTIVES/TASK STATEMENTS

Within this domain, the IS auditor should be able to:

- Evaluate the organization's ability to continue business operations. (T13)
- Evaluate whether IT service management practices align with business requirements. (T20)
- Conduct periodic review of information systems and enterprise architecture. (T21)
- Evaluate IT operations to determine whether they are controlled effectively and continue to support the organization's objectives. (T22)
- Evaluate IT maintenance practices to determine whether they are controlled effectively and continue to support the organization's objectives. (T23)
- Evaluate database management practices. (T24)
- Evaluate data governance policies and practices. (T25)
- Evaluate problem and incident management policies and practices. (T26)
- Evaluate change, configuration, release, and patch management policies and practices. (T27)
- Evaluate end-user computing to determine whether the processes are effectively controlled. (T28)
- Evaluate policies and practices related to asset life cycle management. (T33)

SUGGESTED RESOURCES FOR FURTHER STUDY

Hobbs, Martyn; IT Asset Management: *A Pocket Survival Guide*, IT Governance Publishing, USA, 2011.

ISACA, COBIT® 2019, USA, 2018, *www.isaca.org/cobit*

International Organization for Standardization/International Electrotechnical Commission, ISO/IEC 20000-1:2018, *Information technology—Service*

management—Part 1: Service management system requirements, Switzerland, 2018, *www.iso.org/standard/70636.html*

Mullins, Craig S.; *Database Administration: The Complete Guide to DBA Practices and Procedures, 2nd Edition*, Addison-Wesley Professional, USA, 2012

Snedaker, Susan; *Business Continuity & Disaster Recovery for IT Professionals 2nd Edition*, Syngress Publishing Inc., USA, 2013

Wallace, Michael; Lawrence Webber; *The Disaster Recovery Handbook; A Step-by-Step Plan to Ensure Business Continuity and Protect Vital Operations, Facilities, and Assets, 2nd Edition*, AMACOM, USA, 2010

SELF-ASSESSMENT QUESTIONS

CISA self-assessment questions support the content in this manual and provide an understanding of the type and structure of questions that typically appear on the exam. Often a question will require the candidate to choose the **MOST** likely or **BEST** answer among the options provided. Please note that these questions are not actual or retired exam items. Please see the section "About This Manual" for more guidance regarding practice questions.

4-1 Which one of the following provides the **BEST** method for determining the level of performance provided by similar information processing facility environments?

A. User satisfaction
B. Goal accomplishment
C. Benchmarking
D. Capacity and growth planning

4-2 For mission critical systems with a low tolerance to interruption and a high cost of recovery, the IS auditor, in principle, recommends the use of which of the following recovery options?

A. Mobile site
B. Warm site
C. Cold site
D. Hot site

4-3 Which of the following is the **MOST** effective method for an IS auditor to use in testing the program change management process?

A. Trace from system-generated information to the change management documentation
B. Examine change management documentation for evidence of accuracy
C. Trace from the change management documentation to a system-generated audit trail
D. Examine change management documentation for evidence of completeness

4-4 Which of the following would allow an enterprise to extend its intranet across the Internet to its business partners?

A. Virtual private network
B. Client-server
C. Dial-up access
D. Network service provider

4-5 The classification based on criticality of a software application as part of an IS business continuity plan is determined by the:

A. nature of the business and the value of the application to the business.
B. replacement cost of the application.
C. vendor support available for the application.
D. associated threats and vulnerabilities of the application.

4-6 When conducting an audit of client-server database security, the IS auditor should be **MOST** concerned about the availability of:

A. system utilities.
B. application program generators.
C. systems security documentation.
D. access to stored procedures.

4-7 When reviewing a network used for Internet communications, an IS auditor will **FIRST** examine the:

A. validity of password change occurrences.
B. architecture of the client-server application.
C. network architecture and design.
D. firewall protection and proxy servers.

4-8 An IS auditor should be involved in:

A. observing tests of the disaster recovery plan.
B. developing the disaster recovery plan.
C. maintaining the disaster recovery plan.
D. reviewing the disaster recovery requirements of supplier contracts.

4-9 Data mirroring should be implemented as a recovery strategy when:

A. recovery point objective (RPO) is low.
B. recovery point objective (RPO) is high.
C. recovery time objective (RTO) is high.
D. disaster tolerance is high.

4-10 Which of the following components of a business continuity plan is **PRIMARILY** the responsibility of an organization's IS department?

A. Developing the business continuity plan

B. Selecting and approving the recovery strategies used in the business continuity plan
C. Declaring a disaster
D. Restoring the IT systems and data after a disaster

ANSWERS TO SELF-ASSESSMENT QUESTIONS

4-1 A. User satisfaction is the measure to ensure that an effective information processing operation meets user requirements.
B. Goal accomplishment evaluates effectiveness involved in comparing performance with predefined goals.
C. Benchmarking provides a means of determining the level of performance offered by similar information processing facility environments.
D. Capacity and growth planning are essential due to the importance of IT in organizations and the constant change in technology.

4-2 A. Mobile sites are specially designed trailers that can be quickly transported to a business location or to an alternate site to provide a ready-conditioned information processing facility (IPF).
B. Warm sites are partially configured, usually with network connections and selected peripheral equipment—such as disk drives, tape drives and controllers—but without the main computer.
C. Cold sites have only the basic environment to operate an IPF. Cold sites are ready to receive equipment, but do not offer any components at the site in advance of the need.
D. Hot sites are fully configured and ready to operate within several hours or, in some cases, even minutes.

4-3 **A. When testing change management, the IS auditor should always start with system-generated information, containing the date and time a module was last updated, and trace from there to the documentation authorizing the change.**
B. Focusing exclusively on the accuracy of the documentation examined does not ensure that all changes were, in fact,

documented.

C. To trace in the opposite direction would run the risk of not detecting undocumented changes.

D. Focusing exclusively on the completeness of the documentation examined does not ensure that all changes were, in fact, documented.

4-4 **A. Virtual private network (VPN) technology allows external partners to securely participate in the extranet using public networks as a transport or shared private network. Because of low cost, using public networks (Internet) as a transport is the principal method. VPNs rely on tunneling/encapsulation techniques, which allow the Internet Protocol (IP) to carry a variety of different protocols (e.g., SNA and IPX).**

B. Client-server does not address extending the network to business partners (i.e., client-servers refers to a group of computers within an organization connected by a communications network where the client is the requesting machine and the server is the supplying machine).

C. Although it may be technically possible for an enterprise to extend its intranet using dial-up access, it would not be practical or cost effective to do so.

D. A network service provider may provide services to a shared private network by providing Internet services, but it does not extend an organization's intranet.

4-5 **A. The criticality classification is determined by the role of the application syste**m in supporting the strategy of the organization.

B. The replacement cost of the application does not reflect the relative value of the application to the business.

C. Vendor support is not a relevant factor for determining the criticality classification.

D. The associated threats and vulnerabilities will get evaluated only if the application is critical to the business.

4-6 **A. System utilities may enable unauthorized changes to be made**

to data on the client-server database. In an audit of database security, the controls over such utilities would be the primary concern of the IS auditor.

B. Application program generators are an intrinsic part of client-server technology, and the IS auditor would evaluate the controls over the generators access rights to the database rather than their availability.

C. Security documentation should be restricted to authorized security staff, but this is not a primary concern.

D. Access to stored procedures is not a primary concern.

4-7 A. Reviewing validity of password changes would be performed as part of substantive testing.

B. Understanding the network architecture and design is the starting point for identifying the various layers of information and the access architecture across the various layers, such as client-server applications.

C. The first step in auditing a network is to understand the network architecture and design. Understanding the network architecture and design provides an overall picture of the network and its connectivity.

D. Understanding the network architecture and design is the starting point for identifying the various layers of information and the access architecture across the various layers, such as proxy servers and firewalls.

4-8 **A. The IS auditor should always be present when disaster recovery plans are tested to ensure that the tested recovery procedures meet the required targets for restoration, that recovery procedures are effective and efficient, and to report on the results, as appropriate.**

B. IS auditors may be involved in overseeing plan development, but they are unlikely to be involved in the actual development process.

C. Similarly, an audit of plan maintenance procedures may be conducted, but the IS auditor normally would not have any

responsibility for the actual maintenance.

D. An IS auditor may be asked to comment upon various elements of a supplier contract, but, again, this is not always the case.

4-9 **A. Recovery point objective (RPO) is the earliest point in time at which it is acceptable to recover the data. In other words, RPO indicates the age of the recovered data (i.e., how long ago the data were backed up or otherwise replicated). If RPO is very low, such as minutes, it means that the organization cannot afford to lose even a few minutes of data. In such cases, data mirroring (synchronous data replication) should be used as a recovery strategy.**

B. If RPO is high, such as hours, then other backup procedures—such as tape backup and recovery—could be used.

C. A high recovery time objective (RTO) means that the IT system may not be needed immediately after the disaster declaration/disruption (i.e., it can be recovered later).

D. RTO is the time from the disruption/declaration of disaster during which the business can tolerate nonavailability of IT facilities. If RTO is high, slower recovery strategies that bring up IT systems and facilities can be used.

4-10 A. Members of the organization's senior management are primarily responsible for overseeing the development of the business continuity plan for an organization and are accountable for the results.

B. Management is also accountable for selecting and approving the strategies used for disaster recovery.

C. IT may be involved in declaring a disaster but is not primarily responsible.

D. The correct choice is restoring the IT systems and data after a disaster. The IT department of an organization is primarily responsible for restoring the IT systems and data after a disaster within the designated timeframes.

PART A: INFORMATION SYSTEMS OPERATIONS

4.0 INTRODUCTION

IT service management practices are important to provide assurance to users and to management that the expected level of service will be delivered. Service level expectations are derived from the organization's business objectives. IT service delivery includes IS operations, IT services, and management of IS and the groups responsible for supporting them. IT services are built on service management frameworks.

4.1 COMMON TECHNOLOGY COMPONENTS

This section introduces:
- Technology components
- Hardware platforms
- Basic concepts of, and history behind, the different types of computers
- Advances in IT

Also discussed are the key audit considerations, such as capacity management, system monitoring, maintenance of hardware and typical steps in the acquisition of new hardware.

> **Note:** Vendor-specific terminology is used within this manual for illustrative purposes only. Candidates will not be examined on the components of vendor-specific hardware offerings or on vendor-specific terminology unless this terminology has become generalized and is used globally.

4.1.1 COMPUTER HARDWARE COMPONENTS AND

ARCHITECTURES

Computer-system hardware components are interdependent components that perform specific functions and can be classified as either processing or input/output.

Processing Components

The central component of a computer is the central processing unit (CPU). Computers may:

- Have the CPU on a single chip (microprocessors)
- Have more than one CPU (multi-processor)
- Contain multiple CPUs on a single chip (multi-core processors)

The CPU consists of an arithmetic logic unit (ALU), a control unit and an internal memory. The control unit contains electrical circuits that control/direct all operations in the computer system. The ALU performs mathematical and logical operations. The internal memory (i.e., CPU registers) is used for processing transactions.

Other key components of a computer include:

- Motherboard
- Random access memory (RAM)
- Read-only memory (ROM)
- Permanent storage devices (hard disk drive or solid-state drive [SSD])
- A power supply unity

An SSD is nonvolatile storage device that stores persistent data on solid-state flash memory. SSDs have no moving components and, therefore, require less energy. This distinguishes them from hard disk drives, which contain spinning disks and movable read/write heads.

Input/Output Components

The input/output (I/O) components are used to pass instructions/information to the computer and to display or record the output generated by the computer. Some components, such as the keyboard and mouse, are input-only devices, while others, such as the touch screen, are both input and output devices. Printers are an example of an output-only device.

Types of Computers

Computers can be categorized according to several criteria—mainly their processing power, size and architecture. These categories are shown in **figure 4.1**.

4.1.2 COMMON ENTERPRISE BACK-END DEVICES

In a distributed environment, many different devices are used to deliver application services. One factor that has significantly changed in recent years is the rapid growth of the Internet of Things (IoT). Organizations need to know and embrace the many connected items in use, including cars, thermostats, video cameras, mattresses and medical equipment, and understand how they are affecting operations. Although increased innovation, productivity and services offer benefits, IoT use also risks data leakage and privacy issues, among others. See chapter 5, Protection of Information Assets, for more information.

Following are some of the most common devices encountered:

- **Print servers**—Businesses of all sizes require that printing capability be made available to users across multiple sites and domains. Generally, a network printer is configured based on where the printer is physically located and who within the organization needs to use it. Print servers allow businesses to consolidate printing resources for cost savings.
- **File servers**—File servers provide for organizationwide access to files and programs. Document repositories can be centralized to a few locations within the organization and controlled with an access-control matrix. Group collaboration and document management are easier when a document repository is used, rather than dispersed storage across multiple workstations.
- **Application (program) servers**—Application servers typically host the software programs that provide application access to client computers, including processing the application business logic and communication with the application's database. Consolidation of applications and licenses in servers enables centralized management and a more secure environment.
- **Web servers**—Web servers provide information and services to external customers and internal employees through web pages. They are normally

accessed by their uniform resource locators (URLs).

Figure 4.1—Common Types of Computers	
Supercomputers	Very large and expensive computers with the highest processing speed, designed to be used for specialized purposes or fields that require extensive processing power (e.g., complex mathematical or logical calculations). They are typically dedicated to a few specific specialized system or application programs.
Mainframes	Large, general-purpose computers that are made to share their processing power and facilities with thousands of internal or external users. Mainframes accomplish this by executing a large variety of tasks almost simultaneously. The range of capabilities of these computers is extensive. A mainframe computer often has its own proprietary OS that can support background (batch) and real-time (online) programs operating parallel applications. Mainframes have traditionally been the main data processing and data warehousing resource of large organizations and, as such, have long been protected by a number of the early security and control tools.
High-end and midrange servers	Multiprocessing systems capable of supporting thousands of simultaneous users. In size and power, they can be comparable to a mainframe. High-end/midrange servers have many of the control features of mainframes such as online memory and CPU management, physical and logical partitioning, etc. Their capabilities are also comparable to mainframes in terms of speed for processing data and execution of client programs, but they cost much less than mainframes. Their OSs and system software base components are often commercial products. The higher-end devices generally use UNIX and, in many cases, are used as database servers while smaller devices are more likely to utilize the Windows OS and be used as application servers and file/print servers.
Personal computers (PCs)	Small computer systems referred to as PCs or workstations that are designed for individual users, inexpensively priced and based on microprocessor technology. Their use includes office automation functions such as word processing, spreadsheets and email; small database management; interaction with web-based applications; and others such as personal graphics, voice, imaging, design, web access and entertainment. Although designed as single-user systems, these computers are commonly linked together to form a network.
Thin client computers	These are personal computers that are generally configured with minimal hardware features (e.g., diskless workstation) with the intent being that most processing occurs at the server level using software, such as Microsoft Terminal Services or Citrix Presentation Server, to access a suite of applications.
Laptop computers	Lightweight (under 10 pounds/5 kilograms) personal computers that are easily transportable and are powered by a normal AC connection or by a rechargeable battery pack. Similar to the desktop variety of personal computers in capability, they have similar CPUs, memory capacity and disk storage capacity, but the battery pack makes them less vulnerable to power failures.

	Being portable, these are vulnerable to theft. Devices may be stolen to obtain information contained therein and hijack connectivity, either within an internal local area network (LAN) or remotely.
Smartphones, tablets and other handheld devices	Handheld devices that enable their users to use a small computing device as a substitute for a laptop computer. Some of its uses include a scheduler, a telephone and address book, creating and tracking to-do lists, an expense manager, eReader, web browser, and an assortment of other functions. Such devices can also combine computing, telephone/fax and networking features together so they can be used anytime and anywhere. Handheld devices are also capable of interfacing with PCs to back up or transfer important information. Likewise, information from a PC can be downloaded to a handheld device.

- **Proxy servers**—Proxy servers provide an intermediate link between users and resources. As opposed to direct access, proxy servers will access services on a user's behalf. Depending on the services being proxied, a proxy server may render more secure and faster response than direct access.
- **Database servers**—Database servers store data and act as a repository. The servers concentrate on storing information rather than presenting it to be usable. Application servers and web servers use the data stored in database servers and process the data into usable information.
- **Appliances (specialized devices)**—Appliances provide a specific service and normally are not capable of running other services. As a result, the devices are significantly smaller and faster, and very efficient. Capacity and performance demands require certain services to be run on appliances instead of generic servers. Examples of appliances are:
  - Firewalls
  - Intrusion detection systems (IDSs)
  - Intrusion prevention systems (IPSs)
  - Switches
  - Routers
  - Virtual private networks (VPNs)
  - Load balancers

Note: See chapter 5, Protection of Information Assets, for more information on these appliances.

4.1.3 UNIVERSAL SERIAL BUS

The universal serial bus (USB) is a serial bus standard that interfaces devices with a host. USB was designed to allow connection of many peripherals to a single standardized interface socket and to improve the plug-and-play capabilities by allowing hot swapping or allowing devices to be connected and disconnected without rebooting the computer or turning off the device. Other convenient features include providing power to low-consumption devices without the need for an external power supply and allowing many devices to be used without requiring installation of manufacturer-specific, individual device drivers.

USB ports overcome the limitations of the serial and parallel ports in terms of speed and the actual number of connections that can be made. USB 2.0 specifications support data transfer at up to 480 megabits per second (Mbps). USB 3.0 can transfer data at up to ten times that speed, five gigabits per second (Gbps), and the latest version USB3.1 is capable of transfer speeds up to 10 Gbps.

USB ports can connect computer peripherals, such as mice, keyboards, tablets, gamepads, joysticks, scanners, digital cameras, printers, personal media players, flash drives and external hard drives. Most operating systems (OSs) recognize when a USB device is connected and load the necessary device drivers.

A **memory card** or **flash drive** is a solid-state electronic data storage device that is used with digital cameras, handheld and mobile computers, telephones, music players, video game consoles and other electronics. They offer high recordability, power-free storage, a small form factor and rugged environmental specifications. Examples include Memory Stick, CompactFlash, SD (secure digital) and flash drive.

Risk Related to USBs

Risk related to the use of USBs includes the following:

- **Viruses and other malicious software**—USB drives present a vector for computer viruses that is very difficult to defend against. Whenever files are transferred between two machines, there is a risk that malware (e.g.,

viruses, spyware and keyloggers) will be transmitted, and USB drives are no exception. Some USB drives include a physical switch that can put the drive in read-only mode. When transferring files to an untrusted machine, a USB drive that is in read-only mode will prevent any data (including viruses) to be written to the device.

- **Data theft**—Hackers, corporate spies and disgruntled employees steal data, and in many cases, these are crimes of opportunity. With a USB drive, any unattended and unlocked PC with a USB port provides an opportunity for criminal activity. Social engineering can give a hacker physical access to a corporate PC to steal data or plant spyware.
- **Data and media loss**—The portability of USB drives presents an increased risk for lost data and media. If an unencrypted USB device is lost, any individual who finds the device will be able to access the data on the drive.
- **Corruption of data**—If the drive is improperly unplugged, then data loss can occur due to corruption. USB drives differ from other types of removable media, such as CD-ROM and DVD-ROM devices, because the computer is not automatically alerted when USB drives are removed. Users of USB drives must alert the computer when they intend to remove the device; otherwise, the computer will be unable to perform the necessary clean-up functions required to disconnect the device, especially if files from the device are currently open.
- **Loss of confidentiality**—Because of its convenient small physical size and large logical size, a significant amount of data can be stored on a USB drive. Some stored information is confidential, and loss of data becomes a risk when the drive is lost, increasing the risk of the data falling into the hands of a competitor. Legal issues can also be associated with loss of confidentiality. For example, in the United States, lost or compromised patient data can indicate a breach of patient privacy, thus violating the Health Insurance Portability and Accountability Act (HIPAA).

Security Controls Related to USBs

The following controls can be used to help reduce risk associated with the use of USB devices:

- **Encryption**—An ideal encryption strategy allows data to be stored on the USB drive but renders the data useless without the required encryption key, such as a strong password or biometric data. Products are available to

implement strong encryption and comply with the latest Federal Information Processing Standards (FIPS). Encryption is a good method to protect information written to the device from loss or theft of the device. But unless the information is also encrypted on the network or local workstation hard drive, sensitive data still are exposed to theft.

- **Granular control**—Products are available to provide centralized management of ports. Because management is accomplished via the use of specialized software, centralized management from the enterprise to the individual system is possible. As with all security issues, a technological solution in isolation is insufficient. Strong policies, procedures, standards and guidelines must be put in place to ensure secure operation of memory card and USB drives. Further, an aggressive user awareness program is necessary to effect changes in employee behavior.
- **Security personnel education**—Flash drives are so small and unobtrusive that they are easily concealed and removed from an enterprise. Physical security personnel should understand USB devices and the risk they present.
- **The lock desktop policy enforcement**—In higher-risk environments, desktop computers should be configured to automatically lock after short intervals.
- **Antivirus policy**—Antivirus software should be configured to scan all attached drives and removable media. Users should be trained to scan files before opening them.
- **Use of secure devices only**—Enforce the use of encryption. Software is available to manage USBs, enforcing encryption or only accepting encrypted devices.
- **Inclusion of return information**—If a USB drive is lost or misplaced, including a small, readable text file containing return information may help with device retrieval. It would be prudent to NOT include company details, but rather a phone number or post office box. It also would be prudent to include a legal disclaimer that clearly identifies the information on the drive as confidential and protected by law.

4.1.4 RADIO FREQUENCY IDENTIFICATION

Radio frequency identification (RFID) uses radio waves to identify tagged

objects within a limited radius. A tag consists of a microchip and an antenna. The microchip stores information along with an ID to identify a product, while the antenna transmits the information to an RFID reader.

The power needed to drive the tag can be derived in two modes. The first mode, used in passive tags, draws power from the incidental radiation arriving from the reader. The second and more expensive mode, used in active tags, derives its power from batteries and therefore is capable of using higher frequencies and achieving longer communication distances. An active tag is reusable and can contain more data.

Tags can be used to identify an item based on either direct product identification or carrier identification. In the case of the latter, an article's ID is manually fed into the system (e.g., using a bar code) and is used along with strategically placed radio frequency readers to track and locate the item.

Applications of RFID

Application of RFID include the following:

- **Asset management**—RFID-based asset management systems are used to manage inventory of any item that can be tagged. Asset management systems using RFID technology offer significant advantages over paper-based or bar-code systems, including the ability to read the identifiers of multiple items nearly simultaneously without optical line of sight or physical contact.
- **Tracking**—RFID asset management systems are used to identify the location of an item or, more accurately, the location of the last reader that detected the presence of the tag associated with the item.
- **Authenticity verification**—The tag provides evidence of the source of a tagged item. Authenticity verification often is incorporated into a tracking application.
- **Matching**—Two tagged items are matched with each other and a signal (e.g., a light or tone) is triggered if one of the items is later matched with an incorrect tagged item.
- **Process control**—This allows business processes to use information associated with a tag (or the item attached to the tag) and to take a customized action.

- **Access control**—The system uses RFID to automatically check whether an individual is authorized to physically access a facility (e.g., a gated campus or a specific building) or logically access an information technology system.
- **Supply chain management (SCM)**—SCM involves the monitoring and control of products from manufacture to distribution to retail sale. SCM typically bundles several application types, including asset management, tracking, process control and payment systems.

Risk Associated With RFID

Some of the risk associated with RFID includes:

- **Business process risk**—Direct attacks on RFID system components can undermine the business processes that the RFID system was designed to enable.
- **Business intelligence risk**—An adversary or competitor can gain unauthorized access to RFID-generated information and use the information to harm the interests of the organization implementing the RFID system.
- **Privacy risk**—Personal privacy rights or expectations may be compromised if an RFID system uses what is considered personally identifiable information for a purpose other than originally intended or understood. The personal possession of functioning tags also is a privacy risk because possession can enable tracking of those tagged items.
- **Externality risk**—RFID technology can represent a threat to non-RFID-networked or non-RFID-collocated systems, assets and people. An important characteristic of RFID that impacts the risk is that RF communication is invisible to operators and users.

Security Controls for RFID

Some security controls for RFID include:

- **Management**—A management control involves oversight of the security of the RFID system. For example, management staff of an organization may need to update existing policies to address RFID implementations, such as security controls needed for an RF subsystem.
- **Operational**—An operational control involves the actions performed on a daily basis by the system's administrators and users. For example, RFID

systems need operational controls that ensure the physical security of the systems and their correct use.
- **Technical**—A technical control uses technology to monitor or restrict the actions that can be performed within the system. RFID systems need technical controls for several reasons, such as protecting or encrypting data on tags, causing tags to self-destruct and protecting or encrypting wireless communications.

4.1.5 HARDWARE MAINTENANCE PROGRAM

To ensure proper operation, hardware must be routinely cleaned and serviced. Maintenance requirements vary based on complexity and performance workloads (e.g., processing requirements, terminal access and number of applications running). Maintenance should be scheduled to closely coincide with vendor-provided specifications. Maintenance is also important for environmental hardware that controls temperature and humidity, fire protection and electrical power. The hardware maintenance program is designed to document the performance of this maintenance.

Information typically maintained by this program includes:
- Reputable service company information for each hardware resource requiring routine maintenance
- Maintenance schedule information
- Maintenance cost information
- Maintenance performance history information, such as planned versus unplanned, executed and exceptional

IS management should monitor, identify and document any deviations from vendor maintenance specifications and provide supporting arguments for this deviation.

When performing an audit of this area, the IS auditor should:
- Ensure that a formal maintenance plan has been developed and approved by management and is being followed.
- Identify maintenance costs that exceed budget or are excessive. These overages may be an indication of a lack of adherence to maintenance procedures or of upcoming changes to hardware. Proper inquiry and

follow-up procedures should be performed.

Hardware Monitoring Procedures

The following are typical procedures and reports for monitoring the effective and efficient use of hardware:

- **Availability reports**—These reports indicate the time periods during which the computer is in operation and available for use by users or other processes. A key concern addressed by this report is excessive IS unavailability, referred to as downtime. This unavailability may indicate inadequate hardware facilities, excessive OS maintenance, the need for preventive maintenance, inadequate environmental facilities (e.g., power supply or air conditioning) or inadequate training for operators.
- **Hardware error reports**—These reports identify CPU, I/O, power and storage failures. These reports should be reviewed by IS operations management to ensure that equipment is functioning properly, to detect failures and to initiate corrective action. The IS auditor should be aware that attribution of an error in hardware or software is not necessarily easy and immediate. Reports should be checked for intermittent or recurring problems, which might indicate difficulties in properly diagnosing the errors.
- **Asset management reports**—These reports provide an inventory of network-connected equipment, such as PCs, servers, routers and other devices.
- **Utilization reports**—These automated reports document the use of the machine and peripherals. Software monitors are used to capture utilization measurements for processors, channels and secondary storage media, such as disk and tape drives. Depending on the OS, resource utilization for multiuser computing environments found in mainframe/large-scale computers should average in the 85- to 95-percent range, with allowances for utilization occasionally reaching 100 percent and falling below 70 percent. Trends from utilization reports can be used by IS management to predict whether more or fewer processing resources are required.

4.1.6 HARDWARE REVIEWS

When auditing infrastructure and operations, hardware reviews should

include the areas shown in **figure 4.2**.

4.2 IT ASSET MANAGEMENT

An asset is something of either tangible or intangible value that is worth protecting and includes people, information, infrastructure, finances and reputation. However, an asset cannot be effectively protected or managed if it is not identified. Likewise, it makes it more difficult to protect an asset if its location is unknown or no owner is assigned.

Figure 4.2—Hardware Reviews

Areas to Review	Questions to Consider
Hardware acquisition plan	• Is the plan aligned with business requirements? • Is the plan aligned with the enterprise architecture? • Is the plan compared regularly to business plans to ensure continued synchronization with business requirements? • Is the plan synchronized with IS plans? • Have criteria for the acquisition of hardware been developed? • Is the environment adequate to accommodate the currently installed hardware and new hardware to be added under the approved hardware acquisition plan? • Are the hardware and software specifications, installation requirements and the likely lead time associated with planned acquisitions adequately documented?
Acquisition of hardware	• Is the acquisition in line with the hardware acquisition plan? • Have the IS management staff issued written policy statements regarding the acquisition and use of hardware, and have these statements been communicated to the users? • Have procedures and forms been established to facilitate the acquisition approval process? • Are requests accompanied by a cost-benefit analysis? • Are purchases routed through the purchasing department to streamline the process, avoid duplications, ensure compliance with tendering requirements and legislation and to take advantage of quantity and quality benefits such as volume discounts?
IT asset management	• Has the hardware been tagged? • Has an owner been designated? • Where will the hardware be located? • Have we retained a copy of the contracts/SLAs?
Capacity management and monitoring	• Are criteria used in the hardware performance monitoring plan based on historical data and analysis obtained from the IS trouble

	logs, processing schedules, job accounting system reports, preventive maintenance schedules and reports? • Is continuous review performed of hardware and system software performance and capacity? • Is monitoring adequate for equipment that has been programmed to contact its manufacturer (without manual or human intervention) in the case of equipment failure?
Preventive maintenance schedule	• Is the prescribed maintenance frequency recommended by the respective hardware vendors being observed? • Is maintenance performed during off-peak workload periods? • Is preventive maintenance performed at times other than when the system is processing critical or sensitive applications?
Hardware availability and utilization reports	• Is scheduling adequate to meet workload schedules and user requirements? • Is scheduling sufficiently flexible to accommodate required hardware preventive maintenance? • Are IS resources readily available for critical application programs?
Problem logs Job accounting system reports	• Have IS management staff reviewed hardware malfunctions, reruns, abnormal system terminations and operator actions?

The first step in IT asset management is the process of identifying and creating an inventory of IT assets. The inventory record of each information asset should include:

- Owner
- Designated custodian
- Specific identification of the asset
- Relative value to the organization
- Loss implications and recovery priority
- Location
- Security/risk classification
- Asset group (where the asset forms part of a larger information system)

Common methods to build the initial inventory include consulting the purchasing system, reviewing contracts and reviewing the software currently installed, using tools, such as Microsoft® System Center Configuration Manager, Spiceworks and ManageEngine.

IT asset management is a fundamental prerequisite to developing a meaningful security strategy. Developing a list of assets is the first step in

managing software licenses (see section 4.7.6, Software Licensing Issues) and classifying and protecting information assets (see section 5.1, Information Asset Security Frameworks, Standards and Guidelines).

IT asset management should be employed for software and hardware assets. It is common to physically tag hardware assets.

4.3 JOB SCHEDULING AND PRODUCTION PROCESS AUTOMATION

In complex IS environments, computer systems transfer hundreds to thousands of data files daily. A job schedule is typically created that lists the jobs that must be run and the order in which they are run, including any dependencies. Due to the inherent complexity of this process, automated job scheduling software provides control over the scheduling process. In addition to the scheduling of batch jobs, job scheduling software can be used to schedule tape backups and other maintenance activities. Job scheduling is a major function within the IT department. The schedule includes the jobs that must be run, the sequence of job execution and the conditions that cause program execution. Low-priority jobs can also be scheduled, if time becomes available.

High-priority jobs should be given optimal resource availability, and maintenance functions (such as backup and system reorganization) should, if possible, be performed during nonpeak times. Schedules provide a means of keeping customer demand at a manageable level and permit unexpected or on-request jobs to be processed without unnecessary delay.

Job scheduling procedures are necessary to ensure that IS resources are used optimally, based on processing requirements. Applications are increasingly required to be continually available; therefore, job scheduling (maintenance or long processing times) represents a greater challenge than before.

4.3.1 JOB SCHEDULING SOFTWARE

Job scheduling software is system software used by installations that process a large number of batch routines. The scheduling software sets up daily work

schedules and automatically determines which jobs are to be submitted to the system for processing.

The advantages of using job scheduling software include:
- Job information is set up only once, reducing the probability of an error.
- Job dependencies are defined so that if a job fails, subsequent jobs relying on its output will not be processed.
- Records are maintained of all job successes and failures.
- Security over access to production data can be provided.
- Reliance on operators is reduced.

4.3.2 SCHEDULING REVIEWS

Figure 4.3 describes an audit approach to be considered when reviewing workload job scheduling and personnel scheduling.

Figure 4.3—Scheduling Reviews

Areas to Review	Questions to Consider
• Regularly scheduled applications • Input deadlines • Data preparation time • Estimated processing time • Output deadlines • Procedures for collecting, reporting and analyzing key performance indicators	• Are the items included in SLAs? • Are the items functioning according to the SLAs?
• Job schedule	• Have critical applications been identified and the highest priority assigned to them? • Have processing priorities been established for other applications and are the assigned priorities justified? • Is scheduling of rush/rerun jobs consistent with their assigned priority? • Do scheduling procedures facilitate optimal use of computer resources while meeting service requirements? • Do operators record jobs that are to be processed and the required data files? • Do operators schedule jobs for processing on a predetermined basis and perform them using either automated scheduling software or a manual schedule?
• Daily job schedule	• Is the number of personnel assigned to each shift adequate to

	support the workload? • Does the daily job schedule serve as an audit trail? Does the schedule provide each shift of computer operators with the work to be carried out, the sequence in which programs are to be run and indication when lower-priority work can be performed? • At the end of a shift, does each operator pass to the work scheduler or the next shift of operators a statement of the work completed and the reasons any scheduled work was not finished?
• Console log	• Were jobs run and completed according to the schedule? • If not, are the reasons valid?
• Exception processing logs	• Do operators obtain written or electronic approval from owners when scheduling request-only jobs? • Do operators record all exception processing requests? • Do operators review the exception processing request log to determine the appropriateness of procedures performed?
• Reexecuted jobs	• Are all reexecution of jobs properly authorized and logged for IS management review? • Are procedures established for rerunning jobs to ensure that the correct input files are being used and subsequent jobs in the sequence also are rerun, if appropriate?
• Personnel	• Are personnel who are capable of assigning, changing job schedules or job priorities authorized to do so?

4.4 SYSTEM INTERFACES

A system is a set of elements, including hardware and software, that work together to run one or more computers. System interfaces exist where data output from one application is sent as input to another, with little or no human interaction. Interfaces that involve humans are usually called user interfaces.

System interfaces provide the ability to transfer data even if the systems use different programming languages or were created by different developers. This offers organizations a greater level of flexibility to choose the applications that best serve different areas, even if those areas need to share data.

Generally, data transfers through system interfaces can be sorted into three categories: system-to-system, partner-to-partner and person-to-person.

System-to-system interfaces occur when data is transferred between two systems, whether internal or external. Data may also be transferred to specialized tools for analysis. These uses have increased in part because of the growing popularity of business analytics, which involves transferring data from a repository to an analytic tool to obtain intelligence and insights via data mining.

Partner-to-partner interface occurs when two partners are continuously transferring data back and forth across agreed-upon systems. These transfers are generally done on a regular basis.

Person-to-person transfers are often the most unnoticed and unmanaged. They can be as easy as attaching a data file to an email and sending it. These forms of transfer tend to be more difficult to observe, manage, secure and control.

4.4.1 RISK ASSOCIATED WITH SYSTEM INTERFACES

Recognizing this growth, organizations are focusing more on ensuring that there is a centralized methodology for tracking and managing system interfaces and that there are documentation and audit trails for relevant government regulations. Unmanaged interfaces can add to the risk regarding data security, privacy and error.

It is critical that organizations are able to rely on the integrity of the data exchanged through system interfaces. If an interface is not functioning correctly, one possible consequence is that incorrect management reports (e.g., research, financial, intelligence, performance and competitive) have a significant negative impact on a business and decision-making. Beyond an effect on business value, even a small error can invoke potential legal compliance liability.

4.4.2 SECURITY ISSUES IN SYSTEM INTERFACES

The primary objective of maintaining security of data being transferred through system interfaces is to ensure that the data intended to be extracted from the originating system are the same as the data that were downloaded and recorded in the recipient system. The data need to be protected and

secured throughout the transfer process. The secondary objective is to prevent unauthorized access to the data via interception, malicious activity, error or other means. Unavailability of system interfaces can also affect the reliability of data.

4.4.3 CONTROLS ASSOCIATED WITH SYSTEM INTERFACES

The IS auditor should ensure that the organization has a program that tracks and manages all system interfaces and data transfers, whether internal or external, in line with the business needs and goals. This includes the ability to see all the transfers made, including those that are ad hoc, whether the organization is using a commercial or custom managed file transfer (MFT) system. IS auditors should ensure that the program is able to:

- Manage multiple file transfer mechanisms.
- Use multiple protocols.
- Automatically encrypt, decrypt and electronically sign data files.
- Compress/decompress data files.
- Connect to common database servers.
- Send and retrieve files via email and secure email.
- Automatically schedule regular data transfers.
- Analyze, track and report any attributes of the data being transferred.
- Ensure compliance with appropriate regulatory laws and mandates.
- Offer a checkpoint or restart capability for interruptions.
- Integrate with back-office applications to automate data transfers as much as feasible.

Controls need to be implemented with the objective of ensuring that the data residing on the sending system are precisely the same data that are recorded on the receiving system. For example, an organization may use a software package that can generate controls during the extraction that automatically reconcile the data after they are recorded on the receiving system.

Although automated controls are generally preferred over manual controls, another control can be manual reconciliation by running a report of the data sent and comparing it to a report on the data received. This should be done by

a qualified person who has the ability to detect material differences in the data.

IS auditors should also ascertain if the organization is using encryption, as appropriate for each use, to protect data during the transfer. Encryption is necessary when the risk of unauthorized access or interception is relatively high (e.g., industrial espionage, identity theft, credit card data theft). Additionally, the transfer process may require strong access and authentication controls, and the data files might be password-protected.

There also should be a control over nonrepudiation, which ensures that the intended recipient is the actual recipient of the data.

To ensure that an audit trail is associated with the system interface, the organization needs to capture important information, including who sent the data, when they were sent, when they were received, what data structure (e.g., xls, csv, txt or xml) was used, how the data were sent and who received the data. This includes assessing automated logs of servers along the path, especially if the data are transmitted to an external system where they touch multiple Internet hosts and are more exposed to hackers and cybercriminals.

4.5 END-USER COMPUTING

End users are the people who access business applications that were programmed, serviced and installed by others. End-user computing (EUC) refers to the ability of end users (who typically are not programmers) to design and implement their own application or information system using computer software products. Often, an end-user support manager is a liaison between an IT department and end users.

One of the benefits of EUC is that users can quickly build and deploy applications, taking the pressure off of the IT department. EUC also enables organizations to be more flexible and more rapidly address shifting marketplaces, regulations and consumer interests.

Lack of IT department involvement in EUC also brings associated risk, because the applications may not be subject to an independent review and,

frequently, are not created in the context of a formal development methodology.

This lack of IT department involvement can result in applications that:
- May contain errors and give incorrect results
- Are not subject to change management or release management, resulting in multiple, perhaps different, copies
- Are not secured
- Are not backed up

The lack of IT department oversight of EUC can lead to security risk. Examples include:
- **Authorization**—There may be no secure mechanism to authorize access to the system.
- **Authentication**—There may be no secure mechanism to authenticate users to the system.
- **Audit logging**—This is not available on standard EUC solutions (e.g., Microsoft Excel and Access).
- **Encryption**—The application may contain sensitive data which have not been encrypted or otherwise protected.

In most instances, EUC applications do not pose a significant risk to the enterprise. Nonetheless, management should define risk criteria to determine the criticality of the application. These applications should also be subject to data classification, and those deemed critical enough should be subject to the same controls as any other application.

Organizations need to manage and control EUC and the IS auditor should ensure that policies for the use of EUC exist. An inventory (see section 4.2, IT Asset Management) of all such applications should exist, and those deemed critical enough should be subject to the same controls as any other application.

4.6 DATA GOVERNANCE

With ever-changing data environments—such as the cloud—and data requirements, data maintenance and management are becoming increasingly

complicated. Data also exists in many forms, such as text, numbers, graphics and video. After data are made meaningful, they become information, which is crucial to the operation of an enterprise.

Data governance ensures that:

- Stakeholder needs, conditions and options are evaluated to determine balanced, mutually agreed enterprise objectives to be achieved through the acquisition and management of data/information resources.
- Direction is set for data/information management capabilities through prioritization and decision making.
- Performance and compliance of data/information resources are monitored and evaluated relative to mutually agreed-upon (by all stakeholders) direction and objectives.

Data governance reflects the practice of evaluating requirements and bringing direction and control over data and information so that users have access to that data and can trust and rely on it.

Data governance also involves monitoring the performance of IT operations, specifically those areas that relate to data and its availability, integrity and confidentiality.

4.6.1 DATA MANAGEMENT

The *Data Management Body of Knowledge* (DMBOK) defines data management as “the planning and execution of policies, practices, and projects that acquire, control, protect, deliver, and enhance the value of data and information assets.”

Data management is a component of data architecture, which is a key part of enterprise architecture.

Data Quality

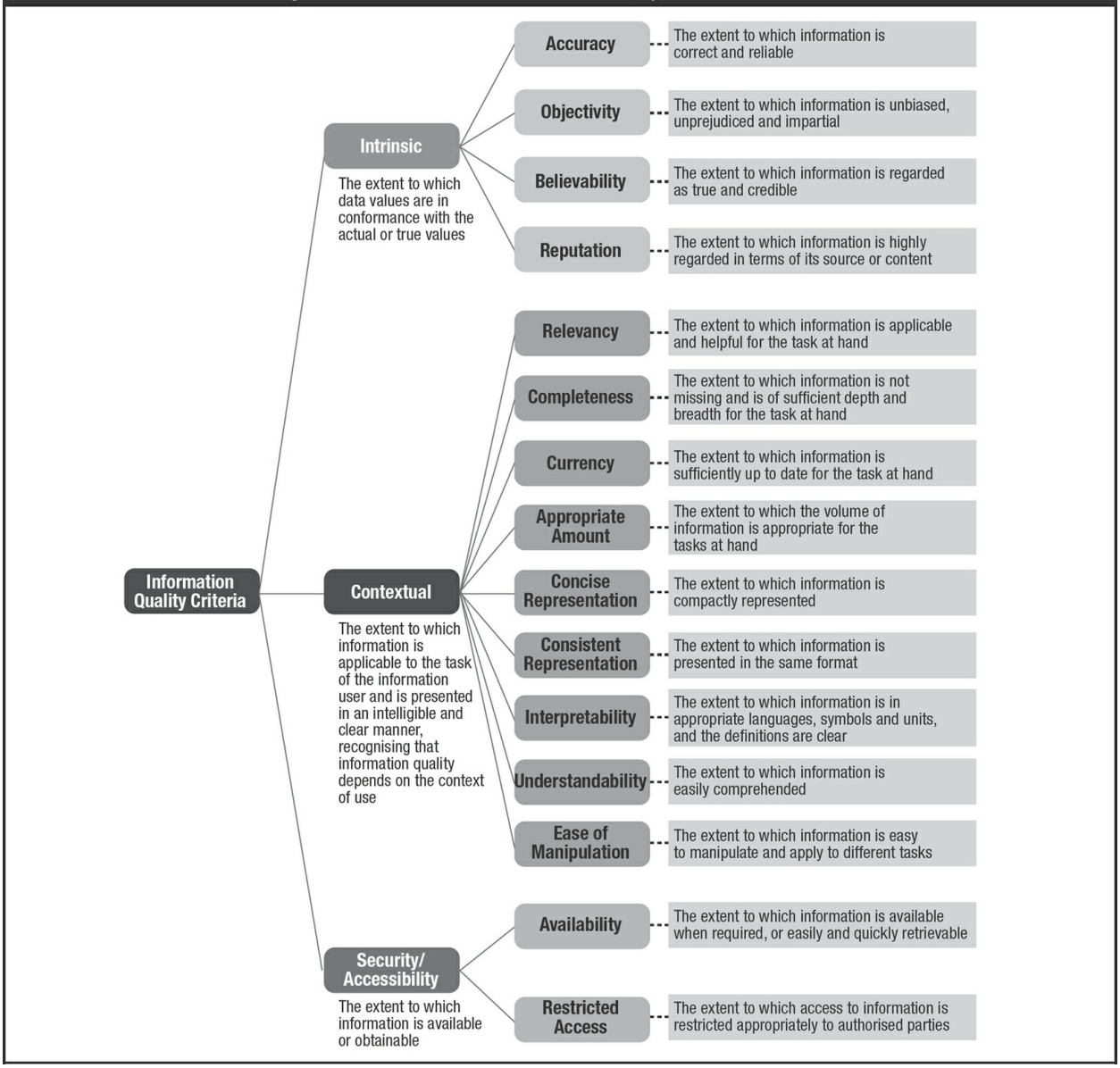
Data quality is key to data management. There are three subdimensions of quality: intrinsic, contextual and security/accessibility. Each subdimension is divided further into several quality criteria, which are defined in **figure 4.4.**

Data Life Cycle

A life cycle describes a series of stages that characterize the course of existence of an organizational investment. Data life cycle management describes the stages that data go through in the course of existence in an organization. The data life cycle includes the following elements:

- **Plan**—The phase in which the creation, acquisition and use of the information resource is prepared. Activities in this phase include understanding information use in the respective business processes, determining the value of the information asset and its associated classification, identifying objectives and planning the information architecture.

Figure 4.4—Information Reference Model: Quality Criteria for Information



Source: ISACA, *COBIT 2019 Framework: Introduction and Methodology*, USA, 2018.

- **Design**—The phase in which more detailed work is done in specifying how the information will look and how systems processing the information will have to work. Activities in this phase may refer to the development of standards and definitions (e.g., data definitions, data collection, access, storage procedures, metadata characteristics and data classification).
- **Build/acquire**—The phase in which the information resource is acquired. Activities in this phase may refer to the creation of data records, the purchase of data and the loading of external files.

- **Use/operate**—This phase includes:
  - Store—The phase in which information is held electronically or in hard copy (or even just in human memory). Activities in this phase may refer to the storage of information in electronic form (e.g., electronic files, databases and data warehouses) or as hard copy (e.g., paper documents).
  - Share—The phase in which information is made available for use through a distribution method. Activities in this phase may refer to the processes involved in getting the information to places where it can be accessed and used (e.g., distributing documents by email). For electronically held information, this life cycle phase may largely overlap with the store phase (e.g., sharing information through database access and file/document servers).
  - Use—The phase in which information is used to accomplish (IT-related and thus enterprise) goals. Activities in this phase may refer to all kinds of information usage (e.g., managerial decision making and running automated processes) and include activities, such as information retrieval and converting information from one form to another. Information use as defined in the information model can be thought of as the purposes for which enterprise stakeholders need information when assuming their roles, fulfilling their activities and interacting with each other.
- **Monitor**—The phase in which it is ensured that the information resource continues to work properly (i.e., to be valuable). Activities in this phase may refer to keeping information up to date and other kinds of information management activities (e.g., enhancing, cleansing, merging and removing duplicate information in data warehouses).
- **Dispose**—The phase in which the information resource is transferred or retained for a defined period, destroyed, or handled as part of an archive as needed. Activities in this phase may refer to information retention, archiving or destroying.

The IS auditor should ensure that the quality of the data allows the organization to meet its strategic objectives. Are the data being captured and processed to required standards? The IS auditor should also ensure that the configuration of the organization's applications and database management systems is in line with organizational objectives. For example, are data being

archived, retained or destroyed in line with a data retention policy?

4.7 SYSTEMS PERFORMANCE MANAGEMENT

Systems performance refers to the study of an entire system, including physical hardware and components and software, and how it operates. Enterprises want to ensure that systems are performing as expected and issues are identified and addressed in a timely fashion. It is important to understand the features of the IS architecture and associated software to aid in the systems performance management process.

4.7.1 IS ARCHITECTURE AND SOFTWARE

The architecture of most computers can be viewed as a number of layers of circuitry and logic, arranged in a hierarchical structure that interacts with the computer's OS. At the base of the hierarchy is the computer hardware, which includes some hard-coded instructions (firmware). The next level up in the hierarchy comprises the nucleus (kernel) functions. Functions of the nucleus relate to basic processes associated with the OS, which include:

- Interrupt handling
- Process creation/destruction
- Process state switching
- Dispatching
- Process synchronization
- Interprocess communication
- Support of I/O processes
- Support of the allocation and reallocation/release of memory

The nucleus is a highly privileged area where access by most users is restricted. Above the nucleus are various OS processes that support users. These processes, referred to as system software, are a collection of computer programs used in the design, processing and control of all computer applications used to operate and maintain the computer system. Comprised of system utilities and programs, the system software ensures the integrity of the system, controls the flow of programs and events in the computer, and manages the interfaces with the computer. Software developed for the computer must be compatible with its OS. Examples include:

- Access control software
- Data communications software
- Database management software
- Program library management systems
- Tape and disk management systems
- Network management software
- Job scheduling software
- Utility programs

Some or all of the above may be built into the OS.

4.7.2 OPERATING SYSTEMS

Before discussion of the various forms of system software, the most significant system software related to a computer—its OS—needs to be further addressed. The OS contains programs that interface between the user, processor and applications software. It is the control program that runs the computer and acts as a scheduler and traffic controller. It provides the primary means of managing the sharing and use of computer resources, such as processors, real memory (e.g., RAM), auxiliary memory (e.g., disk storage) and I/O devices.

Most modern OSs have also expanded the basic OS functionalities to include capabilities for a more efficient operation of system and applications software. For example, all modern OSs possess a virtual storage memory capability that allows programs to use and reference a range of addresses greater than the real memory. This technique of mapping parts of a large slower memory to a faster and smaller working memory is used between various levels of cached memory within modern systems.

OSs vary in the resources managed, comprehensiveness of management and techniques used to manage resources. The type of computer, its intended use, and normal, expected attached devices and networks influence the OS requirements, characteristics and complexity. For example, a single-user microcomputer operating in stand-alone mode needs an OS capable of cataloging files and loading programs to be effective.

A mainframe computer handling large volumes of transactions for consolidation and distribution requires an OS capable of managing extensive resources and many concurrent operations, in terms of application input and output, with a very high degree of reliability. For example, the z/OS operating system from IBM has been engineered specifically to complement this environment.

A server with multiple users interacting with data and programs, from database servers and middleware connections to legacy mainframe applications, requires an OS that can accommodate multiprocessing, multitasking and multithreading. It must be able to share disk space (files) and CPU time among multiple users and system processes and manage connections to devices on the network. For example, the UNIX operating system is designed to specifically address this type of environment.

A microcomputer in a networked environment functioning as a server with specialized functions (e.g., applications, database management systems [DBMSs] and directory/file storage) also can interact with data and programs of multiple users to provide services to client workstations throughout the network.

It is common for OSs to run on virtual servers. In a virtual environment, software is used to partition one physical server into multiple independent virtual servers. Each of these environments can then run its own (and if required different) OS. To the operator, the OS behaves as if it were running on a physical server.

Software Control Features or Parameters

Various OS software products provide parameters and options for the tailoring of the system and activation of features such as activity logging. Parameters are important in determining how a system runs because they allow a standard piece of software to be customized to diverse environments.

Software control parameters deal with:

- Data management
- Resource management

- Job management
- Priority setting

Parameter selections should be appropriate to the organization's workload and control environment structure. The most effective means of determining how controls are functioning within an OS is to review the software control features and/or parameters.

Improper implementation and/or monitoring of OSs can result in undetected errors and corruption of the data being processed and lead to unauthorized access and inaccurate logging of system usage.

Software Integrity Issues

OS integrity is a very important requirement and ability of the OS and involves using specific hardware and software features to:

- Protect itself from deliberate and inadvertent modification
- Ensure that privileged programs cannot be interfered with by user programs
- Provide for effective process isolation to ensure that:
  - Multiple processes running concurrently will not interfere by accident or by design with each other and are protected from writing into each other's memory (e.g., changing instructions, sharing resources, etc.)
  - Enforcement of least privilege where processes have no more privilege than needed to perform functions and modules call on more privileged routines only if, and for as long as, needed.

To maintain system and data integrity, it is necessary to correctly and consistently define, enforce and monitor the operating environment and the granted permissions. IS management is responsible for the implementation of appropriate authorization techniques to prevent nonprivileged users from gaining the ability to execute privileged instructions and thus take control of the entire machine.

For example, IBM mainframe z/OS systems are customized at system generation (SYSGEN) time. When these systems are started (initial program load), important options and parameters are read from information kept in a key system directory (referred to as the SYS1.PARMLIB partitioned data

set). The directory specifies critical initialization parameters that are used to meet the data center's installation requirements (i.e., other system software activated for job scheduling, security, activity logging, etc.). These options, if uncontrolled, provide a nonprivileged user a way to gain access to the OS's supervisory state. The IS auditor should review system configuration directories/files in all OSs for control options used to protect the supervisory state.

Likewise, PC-based client-server Windows, UNIX and Linux OSs have special system configuration files and directories. The existence of program flaws or errors in configuring, controlling and updating the systems to the latest security patches makes them vulnerable to being compromised by perpetrators. Important Windows system options and parameters are set in special system configuration files, referred to as a registry. Therefore, the registry is an important aspect of IS auditing. Noting any changes that take place in the registry is crucial for maintaining the integrity, confidentiality and availability of the systems. In UNIX-based OSs, the same issues are present. Critical system configuration files and directories related to the nucleus (kernel) operations, system start-up, network file sharing and other remote services should be appropriately secured and checked for correctness.

Activity Logging and Reporting Options

Computer processing activity can be logged for analysis of system functions. Following are some of the areas that can be analyzed based on the activity log:

- OS logs to:
  - Identify data file versions used for production processing
  - Evaluate programs scheduled and run
  - Discover utilities or service IDs usage
  - Evaluate OS activities to ensure that the integrity of the OS has not been compromised due to improper changes to system parameters and libraries
- Databases to:
  - Evaluate the efficiency of the database structure
  - Identify database errors/failures
  - Assess database security

- Access control to:
  - Evaluate the access controls over critical data files/bases and programs
  - Evaluate security facilities that are active in communications systems, DBMSs and applications

Many intruders will attempt to alter logs to hide their activities. Secure logging is also needed to preserve evidence authenticity should the logs be required for legal/court use. It is, therefore, important that logs are protected against alteration. A common way to achieve this is to capture, centralize and analyze the logs on a secure server using security information and event management (SIEM) software.

Operating System Reviews

When auditing operating software development, acquisition or maintenance, the details shown in **figure 4.5** should be considered.

4.7.3 ACCESS CONTROL SOFTWARE

Access control software is designed to prevent unauthorized access to data, unauthorized use of system functions and programs, and unauthorized updates/changes to data, and to detect or prevent unauthorized attempts to access computer resources. For more details on access control software, see chapter 5, Protection of Information Assets.

Figure 4.5—Operating Systems Reviews

Areas to Review	Questions to Consider
• System software selection procedures	• Do they align with the enterprise architecture? • Do they comply with short- and long-range IS plans? • Do they meet the IS requirements? • Are they properly aligned with the objectives of the business? • Do they include IS processing and control requirements? • Do they include an overview of the capabilities of the software and control options?
• Feasibility study • Selection process	• Are same selection criteria applied to all proposals? • Has the cost-benefit analysis of system software procedures addressed: – Direct financial costs associated with the product? – Cost of product maintenance?

	– Hardware requirements and capacity of the product? – Training and technical support requirements? – Impact of the product on processing reliability? – Impact on data security? – Financial stability of the vendor's operations?
• System software security	• Have procedures been established to restrict the ability to circumvent logical security access controls? • Have procedures been implemented to limit access to the system interrupt capability? • Have procedures been implemented to manage software patches and keep the system software up-to-date? • Are existing physical and logical security provisions adequate to restrict access to the master consoles? • Were vendor-supplied installation passwords for the system software changed at the time of installation?
• IT asset management	• Has an owner been designated? • Have we retained a copy of the contracts/SLAs? • What is the license agreement? Are we in compliance with it?
• System software implementation	• Are controls adequate in: – Change procedures? – Authorization procedures? – Access security features? – Documentation requirements? – Documentation of system testing? – Audit trails? – Access controls over the software in production?
• Authorization documentation	• Have additions, deletions or changes to access authorization been documented? • Does documentation exist of any attempted violations? If so, has there been follow-up?
• System documentation	• Are the following areas adequately documented: – Installation control statements? – Parameter tables? – Exit definitions? – Activity logs/reports?
• System software maintenance activities	• Is documentation available for changes made to the system software? • Are current versions of the software supported by the vendor? • Is there a defined patching process?
• System software change controls	• Is access to the libraries containing the system software limited to individual(s) needing to have such access? • Are changes to the software adequately documented and tested prior to implementation? • Is software authorized properly prior to moving from the test environment to the production environment?

• Controls over the installation of changed system software	• Have all appropriate levels of software been implemented? • Have predecessor updates taken place? • Are system software changes scheduled for times when the changes least impact IS processing? • Has a written plan been established for testing changes to system software? • Are test procedures adequate to provide reasonable assurance that changes applied to the system correct known problems and that they do not create new problems? • Are tests being completed as planned? • Have problems encountered during testing been resolved and were the changes retested? • Have fallback or restoration procedures been put in place in case of production failure?

4.7.4 DATA COMMUNICATIONS SOFTWARE

Data communications software is used to transmit messages or data from one point to another either locally or remotely. For example, a database request from an end user is transmitted from that user's terminal to an online application, then to a DBMS in the form of messages handled by data communications software. Likewise, responses back to the user are handled in the same manner (i.e., from the DBMS to the online application and back to the user's terminal).

A typical simple data communications system has three components:
1. The transmitter (source)
2. The transmission path (channel or line)
3. The receiver

One-way communication flows in one direction only. In a two-way communication, both ends may simultaneously operate as source and receiver, with data flowing over the same channel in both directions. The data communication system is concerned only with the correct transmission between two points. It does not operate on the content of the information.

A data communication system is divided into multiple functional layers. At each layer, software interfaces with hardware to provide a specific set of functions. All data communication systems have at least a physical layer and a data link layer. (See chapter 5, Protection of Information Assets, for more

information.)

Communication-based applications operate in local area network (LAN) and wide area network (WAN) environments to support:

- Electronic funds transfer (EFT) systems
- Database management systems
- Customer electronic services/electronic data interchange (EDI)
- Internet forums and email The data communication system interfaces with the OS, application programs, database systems, telecommunication address method systems, network control system, job scheduling system and operator consoles.

4.7.5 UTILITY PROGRAMS

Utility programs are system software used to perform maintenance and routines that frequently are required during normal processing operations. Utility programs can be categorized by use, into five functional areas:

1. Understanding application systems (flowcharting software, transaction profile analyzer, executive path analyzer and data dictionary)
2. Assessing or testing data quality (data manipulation utilities, database dump utilities, data comparison utility and query facility)
3. Testing a program's ability to function correctly and maintain data integrity (test data generator, online debugging facility, output analyzer and network simulator)
4. Assisting in faster program development (visual display utility, library copy, text editor, online coding facility, report generators and code generators)
5. Improving operational efficiency (CPU and memory utilization monitors and communication line analyzers)

Smaller computer systems (i.e., PC and server OSs) are often equipped with specific utilities to:

- Operate verification, cleaning and defragmenting of hard disk and removable memory units
- Initialize removable data volumes and volumes of disk/removable memory
- Save/restore system images
- Reconstruct and restore (logically) cancelled files

- Test system units and peripherals

Many of these utility programs can perform outside the security system or can function without producing an audit trail of activity. As a result, access to and use of these sensitive and powerful utilities should be well controlled and restricted.

4.7.6 SOFTWARE LICENSING ISSUES

Software copyright laws must be followed to protect against the possibility of a company paying penalties over copyright infringements and the added reputational risk of being identified as a company that illegally uses software. A software licensing agreement is a contract that establishes the terms and conditions under which a piece of software is being licensed (i.e., made legally available for use) from the software developer (owner) to the user. There are two different software licensing types: free (**figure 4.6**) and paid (**figure 4.7**).

Figure 4.6—Free Software Licensing Types

Type	Description
Open source	The software may be used, copied, studied, modified and redistributed as required. Open source is usually accompanied by the program source and a copy of the software license (for example, the GNU General Public License). A well-known example is Linux.
Freeware	The software is free, but the source code cannot be redistributed. A well-known example is Adobe Acrobat Reader®.
Shareware	The software may be free initially; however, this may only be on a trial basis or have limited functionality compared to the full, commercial version (may also be known as trial version, demo ware or an evaluation copy).

Figure 4.7—Paid Software Licensing Types

Type	Description
Per central processing unit (CPU)	Depends on the power of the server, specifically the number of the CPUs; could include the number of CPU cores

Per seat	Depends on the number of unique users of the system
Concurrent users	Depends on the total number of users using the software within a predefined period of time
Utilization	Depends on how busy the CPU is or the number of users that are active at any one time
Per workstation	Depends on the number of individual workstations (NOT users) that connect to the software
Enterprise	Usually allows unlimited use of the software throughout an organization without the need to apply any of the rules above, although there may be some restrictions

To detect software licensing violations, the IS auditor should:

- Review the listing of all standard, used and licensed application and system software.
- Obtain copies of all software contracts for these to determine the nature of the license agreements, be it an unlimited enterprise license, per-seat license or individual copies.
- Scan the entire network to produce a list of installed software.
- If required, review a list of server specifications including CPUs and cores.
- Compare the license agreements with the software that is actually installed noting any violations.

Options available to prevent software license violations include:

- Ensure a good software asset management process exists (see section 4.2, IT Asset Management).
- Centralize control, distribution and installation of software (includes disabling the ability of users to install software, where possible).
- Require that all PCs be restricted workstations with disabled or locked-down disk drives, USB ports, etc.
- Install metering software on the LAN and require that all PCs access applications through the metered software.
- Regularly scan user networks endpoints to ensure that unauthorized copies of software have not been loaded (achieved by comparing actual software loaded to the list of software assets).
- Enforce documented policies and procedures that require users to sign an agreement not to install software without management authorization and a

software license agreement.

Software licenses are primarily contractual compliance—that is, organizations agree to comply with the terms and conditions of the software publisher, with or without financial consideration. In certain circumstances, an IS auditor may need expert legal opinion to confirm compliance.

Note that some disaster recovery arrangements may require additional licenses and hosting of additional metering software. See section 4.16, Disaster Recovery Plans, for more information.

4.7.7 SOURCE CODE MANAGEMENT

Source code is the language in which a program is written. It is translated into object code by assemblers and compilers and tells the computer what to do. By its very nature, source code may contain intellectual property and should be protected, and access should be restricted.

Organizational access to source code may differ depending on the application and the nature of the agreement with the supplier. If no source code is supplied, it may be important to secure an escrow agreement. If the software is packaged, access to the source code may be granted under license to allow for customized modifications. If the software is bespoke or developed in house, the organization will have full access to the source code. In all instances source code is subject to the software development life cycle (see section 3.3, System Development Methodologies). Source code management is also tightly linked to change management, release management, quality assurance and information security management.

The actual source code should be managed using a version control system (VCS), often called revision control software (RCS). These maintain a central repository, which allows programmers to check out a program source to make changes to it. Checking in the source creates a new revision of the program. A VCS provides the ability to synchronize source changes with changes from other developers, including conflict resolution when changes have been made to the same section of source. A VCS also allows for branching, a copy of the trunk (original main code) that exists independently

to allow for customization for different customers, countries, locations etc.

An example of a popular VCS is Apache™ Subversion®. Git is a distributed version control system (DVCS). While Subversion manages a single centralized repository, a DVCS has multiple repositories. In a DVCS, the entire repository may be replicated locally with changes committed to the master repository when needed. This allows developers to work remotely, without a connection.

The advantages of VCSs include:
- Control of source code access
- Tracking of source code changes
- Allowing for concurrent development
- Allowing rollback to earlier versions
- Allowing for branching

The IS auditor should always be aware of the following:
- Who has access to source code
- Who can commit the code (push the code to production)
- Alignment of program source code to program objects
- Alignment with change and release management
- Backups of source code including those offsite and escrow agreements

4.7.8 CAPACITY MANAGEMENT

Capacity management is the planning and monitoring of computing and network resources to ensure that the available resources are used efficiently and effectively. This requires that the expansion or reduction of resources takes place in parallel with the overall business growth or reduction. The capacity plan should be developed based on input from user and IS management to ensure that business goals are achieved in the most efficient and effective way. This plan should be reviewed and updated at least annually.

Capacity planning should include projections substantiated by experience, considering the growth of existing business and future expansions. The following information is key to the successful completion of this task:

- CPU utilization
- Computer storage utilization
- Telecommunications
- LAN and WAN bandwidth utilization
- I/O channel utilization
- Number of users
- New technologies
- New applications
- Service level agreements (SLAs)

The IS auditor must realize that the amount and distribution of these requirements have intrinsic flexibility. Specialized resources of a given class may have an impact on the requirements for other classes. For example, the proper use of more intelligent terminals may consume less processor power and less communications bandwidth than other terminals. Consequently, the previous information is strictly related to type and quality of used or planned system components.

An element in capacity management is deciding whether to host the organization's applications distributed across several small servers, consolidated onto a few large servers, in the cloud or combinations of the three hosts. Consolidating applications on a few large servers (also known as application stacking) often allows the organization to make better overall use of the resources, but it increases the impact of a server outage, and it affects more applications when the server has to be shut down for maintenance. Using the cloud allows extra capacity to be purchased on demand, but also brings the risk of relying on the supplier.

Larger organizations often have hundreds, if not thousands, of servers that are arrayed in groups referred to as server farms. Where virtual servers are used, these may be organized as private (also known as internal or corporate) clouds.

If an organization has put data storage hardware in place, the IS auditor should review the capacity management plans, which involve data storage utilization and storage area network (SAN) utilization.

Capacity management must also include network devices, such as switches and routers, that comprise physically and logically separated networks (virtual local area networks [VLANs]).

Capacity planning defines the business's requirements for IT capacity, in business and technical terms, and presents the consequences of delivering the required volume of activity through the IT infrastructure and applications—at the right time and with optimal cost. Capacity management ensures that all current and future capacity and performance aspects of the business requirements are provided in a cost-effective manner.

Information system capacity is one of the key business requirements for IT systems. Business operations and processes can only be supported reliably when IT systems provide the required capacity. IT management should understand the capacity requirements prior to the design of their information systems and verify the final design against the capacity requirements. IT management also must monitor capacity on an ongoing basis and provide additional capability as the business grows. For example, a file server may store all business files, but in two years, when the storage reaches the 80 percent threshold, an additional hard disk should be installed to keep up with the storage requirements.

IT capacity—as measured by CPU power and size of memory, hard disk or servers—is expensive. Organizations do not want to acquire more than what they need at the present time. Capacity planning is the process of ensuring that the resource provision can always meet business requirements. By continuously monitoring the threshold of the capacity utilization, additional capacity can be acquired and deployed before it no longer meets business requirements. With capacity management, expensive resources will only be provided when they are needed, thus resulting in a cost savings.

Capacity management monitors resource utilization and helps with resource planning. During procurement of the IT system, the capability management team will work with the architect to estimate resource requirements and to ensure that adequate, but not excessive, resources are provided to support the new solutions. The estimate is normally based on number of transactions, size

of data being stored, transaction processing time and response time, etc. Estimates help determine capability requirements for the new solutions.

Capacity management aims to consistently provide the required IT resources —at the right time and cost and in alignment with current and future requirements of the business. Capacity management increases efficiency and cost savings by deferring the cost of new capacity to a later date and optimizing capacity to business needs. Capacity management reduces the risk of performance problems or failure by monitoring the resource utilization threshold and provision of new resources before a shortage occurs. Capacity management also provides accurate capacity forecasting through application sizing and modeling for new services.

Capacity planning and monitoring includes the elements listed in **figure 4.8**.

Figure 4.8—Capacity Planning and Monitoring Elements	
Development	Develop a capacity plan that describes current and future requirements for capacity of IT resources.
Monitoring	Monitor IT components to ensure that agreed-upon service levels are achieved.
Analysis	Analyze data collected from monitoring activities to identify trends from which normal utilization and service level, or baseline, can be established.
Tuning	Optimize systems for actual or expected workload based on analyzed and interpreted monitoring data.
Implementation	Introduce changes or new capacity to meet new capacity requirements.
Modeling	Model and forecast the behavior of IT resources to determine future capacity trends and requirements.
Application sizing	Take into consideration the predicted resources for new capacity. When designing the application, determine its size (number of concurrent users that can be handled, number of transactions and data storage requirements) and required server capability, memory size, processing power, etc.

4.8 PROBLEM AND INCIDENT MANAGEMENT

Computer resources, like any other organizational asset, should be used in a manner that benefits the entire organization. This includes providing information to authorized personnel when and where it is needed, and at a cost that is identifiable and auditable. Computer resources include hardware, software, telecommunications, networks, applications and data.

Controls over these resources are sometimes referred to as general controls. Effective control over computer resources is critical because of the reliance on computer processing in managing the business.

4.8.1 PROBLEM MANAGEMENT

Problem management aims to resolve issues through the investigation and in-depth analysis of a major incident or several incidents that are similar in nature to identify the root cause. Standard methodologies for root cause analysis include the development of fishbone/Ishikawa cause-and-effect diagrams, brainstorming and the use of the 5 Whys—an iterative question-asking technique used to explore the cause-and-effect relationships underlying a problem.

After a problem is identified and analysis has identified a root cause, the condition becomes a \known error. A workaround can then be developed to address the error state and prevent future occurrences of the related incidents. This problem is then added to the known error database (KEDB). The goal is to proactively prevent reoccurrence of the error elsewhere or, at a minimum, have a workaround that can be provided immediately should the incident reoccur.

Problem management and incident management are related but have different methods and objectives. Problem management’s objective is to reduce the number and/or severity of incidents, while incident management’s objective is to return the affected business process back to its normal state as quickly as possible, minimizing the impact on the business. Effective problem management can show a significant improvement in the quality of service of an IS organization.

4.8.2 PROCESS OF INCIDENT HANDLING

Incident management is one of the critical processes in IT service management (ITSM). See section 4.10, IT Service Level Management, for more information. IT needs to be attended to on a continuous basis to better serve the customer. Incident management focuses on providing increased continuity of service by reducing or removing the adverse effect of disturbances to IT services and covers almost all nonstandard operations of IT services—thereby defining the scope to include any nonstandard event. In addition to initiation, other steps in the incident life cycle include classification, assignment to specialists, resolution and closure.

It is essential for any incident handling process to prioritize items after determining the impact and urgency. For example, there could be a situation where a service request from the chief information officer (CIO) for a printer problem arrives at the same time as a request from the technology team to attend to a server crash. IS management should have parameters in place for assigning the priority of these incidents, considering both the urgency and impact.

Unresolved incidents are escalated based on the criteria set by IS management. Incident management is reactive, and its objective is to respond to and resolve issues restoring normal service (as defined by the SLA) as quickly as possible. Formal SLAs are sometimes in place to define acceptable ranges for various incident management statistics.

4.8.3 DETECTION, DOCUMENTATION, CONTROL, RESOLUTION AND REPORTING OF ABNORMAL CONDITIONS

Because of the highly complex nature of software, hardware and their interrelationships, a mechanism should exist to detect and document any abnormal conditions that could lead to the identification of an error. This documentation generally takes the form of an automated or manual log. See **figures 4.9** and **4.10**.

Figure 4.9—Typical Types of Errors That Are Logged

• Application errors • System errors • Operator errors	• Network errors • Telecommunication errors • Hardware errors

Figure 4.10—Items to Appear in an Error Log Entry	
• Error date • Error resolution description • Error code • Error description • Source of error • Escalation date and time • Initials of the individual responsible for maintaining the log	• Initials of the individual responsible for closing the log entry • Department/center responsible for error resolution • Status code of problem resolution (i.e., problem open, problem closed pending some future specified date, or problem irresolvable in current environment) • Narrative of the error resolution status

For control purposes, the ability to add to the error log should not be restricted. The ability to update the error log, however, should be restricted to authorized individuals, and the updates should be traceable. Proper segregation of duties requires that the ability to close an error log entry be assigned to a different individual than the one responsible for maintaining or initiating the error log entry.

IS management should ensure that the incident and problem management mechanisms are properly maintained and monitored and that outstanding errors are being adequately addressed and resolved in a timely manner.

IS management should develop operations documentation to ensure that procedures exist for the escalation of unresolved problems to a higher level of IS management. While there are many reasons why a problem may remain outstanding for a long period of time, it should not be acceptable for a problem to remain unresolved indefinitely. The primary risk resulting from lack of attention to unresolved problems is the interruption of business operations. An unresolved hardware or software problem could potentially corrupt production data. Problem escalation procedures should be well documented. IS management should ensure that the problem escalation procedures are being adhered to properly. Problem escalation procedures

generally include:

- Names/contact details of individuals who can deal with specific types of problems
- Types of problems that require urgent resolution
- Problems that can wait until normal working hours

Problem resolution should be communicated to appropriate systems, programming, operations and user personnel to ensure that problems are resolved in a timely manner. The IS auditor should examine problem reports and logs to ensure that they are resolved in a timely manner and are assigned to the individuals or groups most capable of resolving the problem.

The departments and positions responsible for problem resolution should be part of problem management documentation. This documentation must be maintained properly to be useful.

4.8.4 SUPPORT/HELP DESK

The responsibility of the technical support function is to provide specialist knowledge of production systems to identify and assist in system change/development and problem resolution. In addition, it is technical support's responsibility to apprise management of current technologies that may benefit overall operations.

Procedures covering the tasks to be performed by the technical support personnel must be established in accordance with an organization's overall strategies and policies. **Figure 4.11** illustrates common support functions.

Figure 4.11—Typical Support Functions
• Determine the source of computer incidents and take appropriate corrective actions. • Initiate problem reports, as required, and ensure that incidents are resolved in a timely manner. • Obtain detailed knowledge of the network, system and applications. • Answer inquiries regarding specific systems. • Provide second- and third-tier support to business user and customer. • Provide technical support for computerized telecommunications processing. • Maintain documentation of vendor software, including issuance of new releases and problem fixes, as well as documentation of utilities and systems developed in house. • Communicate with IS operations to signal abnormal patterns in calls or application behavior.

Support is generally triaged when a help desk ticket/call is initiated and then escalated based on the complexity of the issue and the level of expertise required to resolve the problem.

The primary purpose of the help desk is to service the user. The help desk personnel must ensure that all hardware and software incidents that arise are fully documented and escalated based on the priorities established by management. In many organizations, the help desk function means different things. However, the basic function of the help desk is to be the first, single and central point of contact for users and to follow the incident management process.

4.8.5 NETWORK MANAGEMENT TOOLS

In an organization's modern inter-networking environment, all the tasks in **figure 4.11** can be accomplished by a set of tools generically called network management tools.

Response time reports identify the time necessary for a command entered by a user at a terminal to be answered by the host system. Response time is important because end users experiencing slow response time will be reluctant to utilize IS resources to their fullest extent. These reports typically identify average, worst and best response times over a given time interval for individual telecommunication lines or systems. These reports should be reviewed by IS management and system support personnel to track potential problems. If response time is slow, all possible causes, such as I/O channel bottlenecks, bandwidth utilization and CPU capacity, should be investigated; various solutions should be analyzed; and an appropriate and cost-justified corrective action should be taken.

Downtime reports track the availability of telecommunication lines and circuits. Interruptions due to power/line failure, traffic overload, operator error or other anomalous conditions are identified in a downtime report. If downtime is excessive, IS management should consider the following remedies:

- Add or replace telecommunications lines.
- Switch to a more dependable transmission link (such as dedicated lines

versus shared lines).
- Install backup power supplies.
- Improve access controls.
- Closely monitor line utilization to better forecast user needs, both in the near and long term.

Help desk reports are prepared by the help desk, which is staffed or supported by IT technicians who are trained to handle problems occurring during normal IS usage. If an end user encounters any problem, he/she can contact the help desk for assistance. Help desk facilities are critical to the telecommunication environment since they provide end users with an easy means of identifying and resolving problems quickly, before they have a major impact on IS performance and end-user resource utilization. Reports prepared by the help desk provide a history of the problems and their resolution.

Online monitors check data transmission accuracy and errors. Monitoring can be performed by echo checking (received data are bounced back to sender for verification) and status checking all transmissions, ensuring that messages are not lost or transmitted more than once.

Network monitors provide a real time display of network nodes and status.

Network (protocol) analyzers are diagnostic tools attached to a network link that use network protocols' intelligence for monitoring the packets flowing along the link and produce network usage reports. Network analyzers are typically hardware-based and operate at the data link and/or network level. Output includes the following information:
- Protocol(s) in use
- Type of packets flowing along the monitored link
- Traffic volume analysis
- Hardware errors, noise and software problems
- Other performance statistics (e.g., percentage of used bandwidth)
- Problems and possible solutions

Simple Network Management Protocol (SNMP) is a TCP/IP-based

protocol that monitors and controls variables throughout the network, manages configurations and collects statistics on performance and security. A master console polls all the network devices on a regular basis and displays the global status. SNMP software is capable of accepting, in real-time, specific operator requests. Based on the operator instructions, SNMP software sends specific commands to an SNMP-enabled device and retrieves the required information. To perform all of these tasks, each device (e.g., routers, switches, hubs, PCs and servers) needs to have an SNMP agent running. The SNMP communication occur between all the agents and the console.

4.8.6 PROBLEM MANAGEMENT REPORTING REVIEWS

The audit approach shown in **figure 4.12** should be considered when reviewing problem management reporting.

Figure 4.12—Problem Management Reporting Reviews

Areas to Review	Questions to Consider
• Interviews with IS operations personnel	• Have documented procedures been developed to guide IS operations personnel in logging, analyzing, resolving and escalating problems in a timely manner, in accordance with management's intent and authorization?
• Procedures used by the IT department • Operations documentation	• Are procedures for recording, evaluating, and resolving or escalating any operating or processing problems ad equate? • Are procedures used by the IT department to collect statistics regarding online processing performance adequate and is the analysis accurate and complete? • Are all problems identified by IS operations being recorded for verification and resolution?
• Performance records • Outstanding error log entries • Help desk call logs	• Do problems exist during processing? • Are the reasons for delays in application program processing valid? • Are significant and recurring problems identified, and actions taken to prevent their recurrence? • Were processing problems resolved in a timely manner and was the resolution complete and reasonable? • Are there any reoccurring problems that are not being reported to IS management?

4.9 CHANGE, CONFIGURATION, RELEASE AND PATCH

MANAGEMENT

Change control procedures are a part of the more encompassing function referred to as change management and are established by IS management to control the movement of application changes (programs, jobs, configurations, parameters, etc.) from the test environment, where development and maintenance occur, to the quality assurance (QA) environment, where thorough testing occurs, to the production environment. Typically, IS operations are responsible for ensuring the integrity of the production environment and often serve as the final approvers of any changes to production.

Change management is used when changing hardware, installing or upgrading to new releases of off-the-shelf applications, installing a software patch and configuring various network devices (e.g., firewalls, routers and switches).

The procedures associated with this process ensure that:

- All relevant personnel are informed of the change and when it is happening.
- System, operations and program documentation are complete, up to date and in compliance with the established standards.
- Job preparation, scheduling and operating instructions have been established.
- System and program test results have been reviewed and approved by user and project management.
- Data file conversion, if necessary, has occurred accurately and completely as evidenced by review and approval by user management.
- System conversion has occurred accurately and completely as evidenced by review and approval by user management.
- All aspects of jobs turned over have been tested, reviewed and approved by control/operations personnel.
- Legal or compliance aspects have been considered.
- The risk of adversely affecting the business operation are reviewed and a rollback plan is developed to back out the changes, if necessary.

Apart from change control, standardized methods and procedures for change management are needed to ensure and maintain agreed-on levels in quality service. These methods are aimed at minimizing the adverse impact of any probable incidents triggered by change that may arise.

This is achieved by formalizing and documenting the process of change request, authorization, testing, implementation and communication to the users. Change requests are often categorized into emergency changes, major changes and minor changes, and may have different change management procedures in place for each type of change.

4.9.1 PATCH MANAGEMENT

Patch management is an area of systems management that involves acquiring, testing and installing multiple patches (code changes) to an administered computer system to maintain up-to-date software and often to address security risk. Patch management tasks include the following:

- Maintain current knowledge of available patches.
- Decide what patches are appropriate for particular systems.
- Ensure that patches are installed properly; testing systems after installation.
- Document all associated procedures, such as specific configurations required.

Several products are available to automate patch management tasks. Patches can be ineffective and can cause more problems than they fix. To avoid problems, patch management experts suggest that system administrators take simple steps, such as performing backups and testing patches on non-critical systems prior to installations. Patch management can be viewed as part of change management.

See chapter 3, Information Systems Acquisition, Development and Implementation, for details on program change controls.

4.9.2 RELEASE MANAGEMENT

Software release management is the process through which software is made available to users. The term release is used to describe a collection of authorized changes. The release will typically consist of several problem

fixes and enhancements to the service.

The releases, whether major or minor, will have a unique identity. Sometimes, the minor or small fixes may trigger some other problem. Fully tested, major releases may not have such problems. Because of testing time, space and other constraints, it is also possible to have a partial release, which is known as a delta release. The delta release contains only those items that have undergone changes since the last release.

The releases are controlled, and, if any problems arise in the new release, one should be able to back out completely and restore the system to its previous state. Suitable contingency plans may also be developed, if it is not completely restorable. These plans are developed before the new release is implemented. **Figure 4.13** shows some of the principal types of releases.

Many new system implementations will involve phased delivery of functionality and thus require multiple releases. In addition, planned releases will offer an ongoing process for system enhancement.

The main roles and responsibilities in release management should be defined to ensure that everyone understands their role and level of authority and those of others involved in the process. The organization should decide the most appropriate approach, depending on the size and nature of the systems, the number and frequency of releases required, and any special needs of the users (for example, if a phased rollout is required over an extended period of time). All releases should have a unique identifier that can be used by configuration management.

Planning a release involves:

- Gain consensus on the release's contents.
- Agree to the release strategy (e.g., the phasing over time and by geographical location, business unit and customers).
- Produce a high-level release schedule.

Figure 4.13—Types of Releases

Major releases	Normally contain a significant change or addition of new functionality. A major upgrade or release usually supersedes all preceding minor upgrades. Grouping together several changes facilitates more comprehensive testing and planned user training. Large organizations typically have a predefined timetable for implementing major releases throughout the year (e.g., quarterly). Smaller organizations may have only one release during the year or numerous releases if the organization is quickly growing.
Minor software releases	Upgrades, normally containing small enhancements and fixes. A minor upgrade or release usually supersedes all preceding emergency fixes. Minor releases are generally used to fix small reliability or functionality problems that cannot wait until the next major release. The entire release process should be followed for the preparation and implementation of minor releases, but it is likely to take less time because the development, testing and implementation activities do not require as much time as major releases do.
Emergency software releases	Emergency releases are fixes that require implementation as quickly as possible to prevent significant user downtime to business-critical functions. Depending upon the required urgency of the release, limited testing and release management activities are executed prior to implementation. Such changes should be avoided whenever possible because they increase the risk of errors being introduced.

- Plan resource levels (including staff overtime).
- Agree on roles and responsibilities.
- Produce back-out plans.
- Develop a quality plan for the release.
- Plan acceptance of support groups and the customer.

While change management is the process whereby all changes go through a robust testing and approval process, release management is the process of putting the software changes into production.

4.9.3 IS OPERATIONS

IS operations are processes and activities that support and manage the entire IS infrastructure, systems, applications and data, focusing on day-to-day activities.

IS operations staff is responsible for the accurate and efficient operation of the network, systems and applications and for the delivery of high-quality IS services to business users and customers.

Tasks of the IS operations staff include:
- Execute and monitor scheduled jobs.
- Facilitate timely backup.
- Monitor unauthorized access and use of sensitive data.
- Monitor and review the extent of adherence to IS operations procedures as established by IS and business management.
- Participate in tests of disaster recovery plans (DRPs).
- Monitor the performance, capacity, availability and failure of information resources.
- Facilitate troubleshooting and incident handling.

Procedures detailing instructions for operational tasks and procedures coupled with appropriate IS management oversight are necessary parts of the IS control environment.

This documentation should include:
- Operations procedures that are based on operating instructions and job flows for computer and peripheral equipment
- Procedures for monitoring systems and applications
- Procedures for detecting systems and applications errors and problems
- Procedures for handling IS problems and escalation of unresolved issues
- Procedures for backup and recovery

IS Operations Reviews

Because processing environments vary among installations, a tour of the information processing facility generally provides the IS auditor with a better understanding of operations tasks, procedures and control environment.

Audit procedures should include those shown in **figure 4.14.**

Figure 4.14—IS Operations Reviews

Areas to Review	Questions to Consider
• Observation of IS personnel	• Have controls been put in place to ensure efficiency of operations and adherence to established standards and policies? • Is adequate supervision present? • Have controls been put in place regarding IS management review,

	data integrity and security?
• Operator access	• Is access to files and documentation libraries restricted to operators? • Are responsibilities for the operation of computer and related peripheral equipment limited? • Is access to correcting program and data problems restricted? • Should access to utilities that allow system fixes to software and/or data be restricted? • Is access to production source code and data libraries (including run procedures) limited?
• Operator manuals	• Are instructions adequate to address: – The operation of the computer and its peripheral equipment? – Startup and shutdown procedures? – Actions to be taken in the event of machine/program failure? – Records to be retained? – Routine job duties and restricted activities?
• Access to the library	• Is the librarian prevented from accessing computer hardware? • Does the librarian have access only to the tape management system? • Is access to library facilities provided to authorized staff only? • Is removal of files restricted by production scheduling software? • Does the librarian handle the receipt and return of foreign media entering the library? • Are logs of the sign-in and sign-out of data files and media maintained?
• Contents and location of offline storage	• Are offline file storage media containing production system programs and data clearly marked with their contents? • Are offline library facilities located away from the computer room? • Are policies and procedures adequate for: – Administering the offline library? – Checking out/in media, including requirements for signature authorizations? – Identifying, labeling, delivering and retrieving offsite backup files? – Encryption of offsite backup files (especially if these physically move between locations)? – Inventorying the system for onsite and offsite media, including the specific storage locations of each tape? – Secure disposal/destruction of media, including requirements for signature authorizations?
• File handling procedures	• Have procedures been established to control the receipt and release of files and secondary storage media to/from other locations? • Are internal tape labels used to help ensure that the correct media are mounted for processing? • Are these procedures adequate and in accordance with management’s intent and authorization?

	• Are these procedures being followed?
• Data entry	• Are input documents authorized and do the documents contain appropriate signatures? • Are batch totals reconciled? • Does segregation of duties exist between the person who keys the data and the person who reviews the keyed data for accuracy and errors? • Are control reports being produced? Are the reports accurate? Are the reports maintained and reviewed?
• Lights-out operations	• Remote access to the master console is often granted to standby operators for contingency purposes such as automated software failure. Is access to security sufficient to guard against unauthorized use? • Do contingency plans allow for the proper identification of a disaster in the unattended facility? • Are the automated operation software and manual contingency procedures documented and tested adequately at the recovery site? • Are proper program change controls and access controls present? • Are tests of the software performed on a periodic basis, especially after changes or updates are applied? • Do assurances exist that errors are not hidden by the software and that all errors result in operator notification?

4.10 IT SERVICE LEVEL MANAGEMENT

The fundamental premise associated with ITSM is that IT can be managed through a series of discrete processes that provide service to the business. Although each process area may have separate and distinct characteristics, each process is also highly interdependent with other processes. The processes, after defined, can be better managed through SLAs that serve to maintain and improve customer satisfaction (i.e., with the end business).

ITSM focuses on the business deliverables and covers infrastructure management of IT applications that support and deliver these IT services. This management includes fine tuning IT services to meet the changing demands of the enterprise and measuring and demonstrating improvements in the quality of IT services offered with a reduction in the cost of service in the long term.

IT services can be better managed with an SLA, and the services offered form a basis for such agreements. There is a possibility of a gap between customer

expectations and the services offered, and this is narrowed by the SLA, which completely defines the nature, type, time and other relevant information for the services being offered. SLAs can also be supported by operational level agreements (OLAs), which are internal agreements covering the delivery of services that support the IT organization in its delivery of services.

For example, when a complaint is received, the help desk looks for an available solution from the KEDB after classifying and storing the complaint as an incident. Repeated incidents or major incidents may lead to problems that call for the problem management process. If changes are needed, the change management group of the process/program can provide a supporting role after consulting the configuration management group.

Any required change—whether it originated as a solution to a problem, an enhancement or for any other reason—goes through the change management process. The cost-benefit and feasibility studies are reviewed before the changes are accepted and approved. The risk of the changes should be studied, and a fallback plan should be developed. The change may be for one configuration item or for multiple items, and the change management process invokes the configuration management process.

For example, the software can comprise different systems, each containing different programs and each program having different modules. The configuration can be maintained at the system level, the program level or the module level. The organization may have a policy saying that any changes made at the system level will be released as a new version. It may also decide to release a new version, if it involves changes at the program level for yet another application.

Service management metrics should be captured and appropriately analyzed so that this information can be used to enhance the quality of service. Many organizations have leveraged ITIL and/or ISO 20000 to improve their ITSM.

4.10.1 SERVICE LEVEL AGREEMENTS

An SLA is an agreement between the IT organization and the customer. The SLA details the service(s) to be provided. The IT organization could be an

internal IT department or an external IT service provider, and the customer is the business. The business may acquire IT services from an internal IT organization, such as email services, an intranet, an enterprise resource planning (ERP) system, etc. The business may acquire IT services from an external IT service provider, such as Internet connectivity, hosting of the public website, etc.

The SLA describes the services in nontechnical terms, from the viewpoint of the customer. During the term of the agreement, it serves as the standard for measuring and adjusting the services.

Service-level management is the process of defining, agreeing on, documenting and managing levels of service that are required and cost justified. Service-level management deals with more than the SLAs themselves; it includes the production and maintenance of the service catalog, service review meetings and service improvement plans (SIPs) for areas that are not achieving their SLAs.

The aim of service-level management is to maintain and improve customer satisfaction and to improve the service delivered to the customer. With clear definition of service level, the IT organization or service provider can design the service based on the service level, and the customer can monitor the performance of the IT services. If the services provided do not meet the SLA, the IT organization or service provider must improve the services.

Characteristics of IT services are used to define the SLA. Characteristics that should be considered in the delivery of these services include accuracy, completeness, timeliness and security. Many tools are available to monitor the efficiency and effectiveness of services provided by IT personnel. These tools include:

- **Exception reports**—These automated reports identify all applications that did not successfully complete or otherwise malfunctioned. An excessive number of exceptions may indicate:
  - Poor understanding of business requirements
  - Poor application design, development or testing
  - Inadequate operation instructions

  - Inadequate operations support
  - Inadequate operator training or performance monitoring
  - Inadequate sequencing of tasks
  - Inadequate system configuration
  - Inadequate capacity management
- **System and application log**s—Logs generated from various systems and applications should be reviewed to identify all application problems. These logs provide additional, useful information regarding activities performed on the computer because most abnormal system and application events will generate a record in the logs. Because of the size and complexity of the logs, it is difficult to manually review them. Programs have been developed that analyze the system log and report on specifically defined items. Using this software, the auditor can carry out tests to ensure that:
  - Only approved programs access sensitive data.
  - Only authorized IT personnel access sensitive data.
  - Software utilities that can alter data files and program libraries are used only for authorized purposes.
  - Approved programs are run only when scheduled and, conversely, unauthorized runs do not take place.
  - The correct data file generation is accessed for production purposes.
  - Data files are adequately protected.
- **Operator problem reports**—These manual reports are used by operators to log computer operations problems and their resolutions. Operator responses should be reviewed by IS management to determine whether operator actions were appropriate or whether additional training should be provided to operators.
- **Operator work schedules**—These schedules are generally maintained manually by IS management to assist in human resource planning. By ensuring proper staffing of operation support personnel, IS management is assured that service requirements of end users will be met. This is especially important during critical or heavy computer usage periods. These schedules should be flexible enough to allow for proper cross-training and emergency staffing requirements.

Many IT departments define the level of service that they will guarantee to users of the IT services. This level of service is often documented in SLAs. It

is particularly important to define service levels where there is a contractual relationship between the IT department and the end user or customer. SLAs are often tied to chargeback systems, in which a certain percentage of the cost is apportioned from the end-user department to the IT department. When functions of the IT department are performed by a third party, it is important to have an outsourcing SLA.

Service levels are often defined to include hardware and software performance targets (such as user response time and hardware availability) but can also include a wide range of other performance measures. Such measures might include financial performance measures (e.g., year-to-year incremental cost reduction), human resources measures (e.g., resource planning, staff turnover, development or training) or risk management measures (e.g., compliance with control objectives). The IS auditor should be aware of the different types of measures available and should ensure that they are comprehensive and include risk, security and control measures as well as efficiency and effectiveness measures.

4.10.2 MONITORING OF SERVICE LEVELS

Defined service levels must be regularly monitored by an appropriate level of management to ensure that the objectives of IS operations are achieved. It is also important to review the impact on the customers and other stakeholders of the organization.

For example, a bank may be monitoring the performance and availability of its automated teller machines (ATMs). One of the metrics may be availability of ATM services at expected levels (99.9 percent); however, it may also be appropriate to monitor the impact on customer satisfaction due to nonavailability. Similar metrics may be defined for other services, such as email and Internet.

Monitoring of service levels is essential for outsourced services, particularly if the third party is involved in directly providing services to an organization’s customers. Failure to achieve service levels will have more of an impact on the organization than on the third party. For example, a fraud due to control weakness at a third party may result in reputation loss for the

organization.

It is important to note that when service delivery is outsourced, only responsibility for serviced provision is outsourced—accountability is not and still rests with the organization. If the organization outsources service delivery, the IS auditor should determine how management gains assurance that the controls at the third party are properly designed and operating effectively. Several techniques can be used by management, including questionnaires, onsite visits or an independent third-party assurance report, such as a Statement on Standards for Attestation Engagements 18 (SSAE 18) Service Organization Control (SOC) 1 report or AT-101 (SOC 2 and SOC 3) report.

4.10.3 SERVICE LEVELS AND ENTERPRISE ARCHITECTURE

Defining and implementing an enterprise architecture (EA) helps an organization in aligning service delivery (see section 2.5, Enterprise Architecture, for more information). Organizations may use multiple service delivery channels, such as mobile apps, the Internet, service outlets, third-party service providers and automated kiosks. These channels use different technologies that are serviced by the same backend database.

When considering availability and recovery options, EA best helps in aligning operational requirements that can address the service delivery objectives. For example, an unacceptable recovery time may lead in choosing fault-tolerant, high-availability architecture for critical service delivery channels (see section 4.16.3, Recovery Alternatives, for more information).

4.11 DATABASE MANAGEMENT

DBMS software aids in organizing, controlling and using the data needed by application programs. A DBMS provides the facility to create and maintain a well-organized database. Primary functions include reduced data redundancy, decreased access time and basic security over sensitive data.

DBMS data are organized in multilevel schemes, with basic data elements,

such as the fields (e.g., Social Security number) at the lowest level. The levels above each field have differing properties depending on the architecture of the database.

The DBMS can include a data dictionary that identifies the fields, their characteristics and their use. Active data dictionaries require entries for all data elements and assist application processing of data elements, such as providing validation characteristics or print formats. Passive dictionaries are only a repository of information that can be viewed or printed.

A DBMS can control user access at the following levels:
- User and the database
- Program and the database
- Transaction and the database
- Program and data field
- User and transaction
- User and data field

Some of the advantages of a DBMS include:
- Data independence for application systems
- Ease of support and flexibility in meeting changing data requirements
- Transaction processing efficiency
- Reduction of data redundancy
- Ability to maximize data consistency
- Ability to minimize maintenance cost through data sharing
- Opportunity to enforce data/programming standards
- Opportunity to enforce data security
- Availability of stored data integrity checks
- Facilitation of terminal users' ad hoc access to data, especially through designed query language/application generators

4.11.1 DBMS ARCHITECTURE

Data elements required to define a database are called metadata. This includes data about data elements used to define logical and physical fields, files, data relationships, queries, etc. There are three types of metadata: conceptual schema, external schema and internal schema. If the schemas are

not adjusted to smoothly work together, the DBMS may not be adequate to meet the users' needs.

Detailed DBMS Metadata Architecture

Within each level, there is a data definition language (DDL) component for creating the schema representation necessary for interpreting and responding to the user's request. At the external level, a DBMS will typically accommodate multiple DDLs for several application programming languages compatible with the DBMS. The conceptual level will provide appropriate mappings between the external and internal schemas. External schemas are location independent of the internal schema.

Data Dictionary/Directory System

A data dictionary/directory system (DD/DS) helps define and store source and object forms of all data definitions for external schemas, conceptual schemas, the internal schema and all associated mappings. The data dictionary contains an index and description of all the items stored in the database. The directory describes the location of the data and the access method.

DD/DS provides the following functional capabilities:

- A data definition language processor, which allows the database administrator to create or modify a data definition for mappings between external and conceptual schemas
- Validation of the definition provided to ensure the integrity of the metadata
- Prevention of unauthorized access to, or manipulation of, the metadata
- Interrogation and reporting facilities that allow the DBA to make inquiries on the data definition

DD/DS can be used by several DBMSs; therefore, using one DD/DS could reduce the impact of changing from one DBMS to another DBMS. Some of the benefits of using DD/DS include:

- Enhance documentation.
- Provide common validation criteria.
- Facilitate programming by reducing the needs for data definition.
- Standardize programming methods.

4.11.2 DATABASE STRUCTURE

There are three major types of database structure: hierarchical, network and relational. Most DBMSs have internal security features that interface with the OS access control mechanism/package. A combination of the DBMS security features and security package functions is often used to cover all required security functions. Types of DBMS structures are discussed in the following paragraphs.

Hierarchical database model—In this model there is a hierarchy of parent and child data segments. To create links between them, this model uses parent-child relationships. These are 1:N (one-to-many) mappings between record types represented by logical trees, as shown in **figure 4.15**. A child segment is restricted to having only one parent segment, so data duplication is necessary to express relationships to multiple parents. Subordinate segments are retrieved through the parent segment. Reverse pointers are not allowed. When the data relationships are hierarchical, the database is easy to implement, modify and search. The registry in Microsoft Windows is an example of a hierarchical database. They are also used in geographic information systems.

Network database model—In the network model, the basic data modeling construct is called a set. A set is formed by an owner record type, a member record type and a name. A member record type can have that role in more than one set, so a multiowner relationship is allowed. An owner record type can also be a member or owner in another set. Usually, a set defines a 1:N relationship, although one-to-one (1:1) is permitted. A disadvantage of the network model is that such structures can be extremely complex and difficult to comprehend, modify or reconstruct in case of failure. This model is rarely used in current environments. See **figure 4.16**. The hierarchical and network models do not support high-level queries. The user programs must navigate the data structures.

Relational database model—An example of a relational database can be seen in **figure 4.17**. The relational model is based on the set theory and relational calculations. A relational database allows the definition of data structures, storage/retrieval operations and integrity constraints. In such a

database, the data and relationships among these data are organized in tables. A table is a collection of rows, also known as tuples, and each tuple in a table contains the same columns. Columns, called domains or attributes, correspond to fields. Tuples are equal to records in a conventional file structure. Relational databases are used in most common ERP Systems. Common relational database management systems (RDBMS) include Oracle®, IBM® DB2® and Microsoft SQL Server.

Relational tables have the following properties:

- Values are atomic, i.e., a single unit that is irreducible
- Each row is uniquely identifiable.
- Column values are of the same kind.

Figure 4.15—Organization of a Hierarchical Database

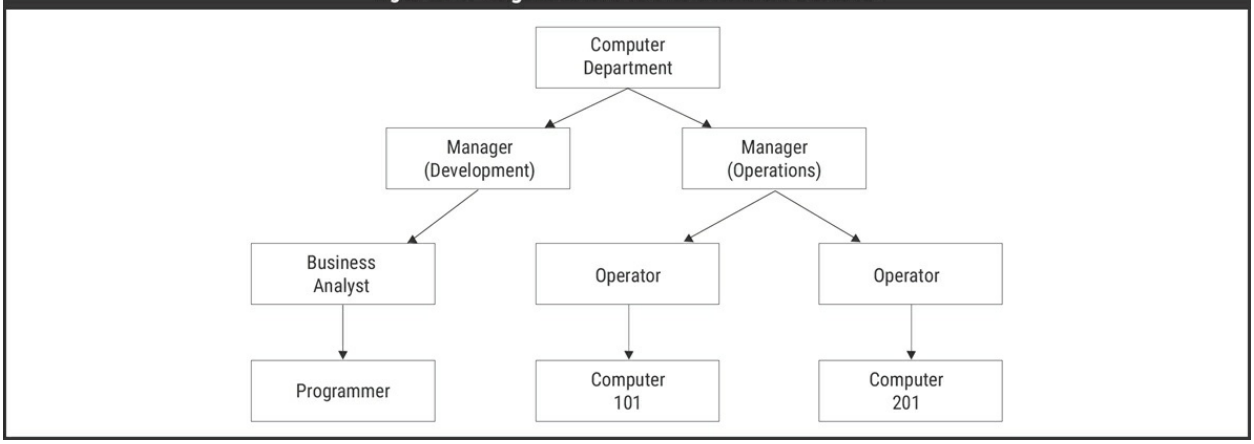


Figure 4.16— Organization of a Network Database

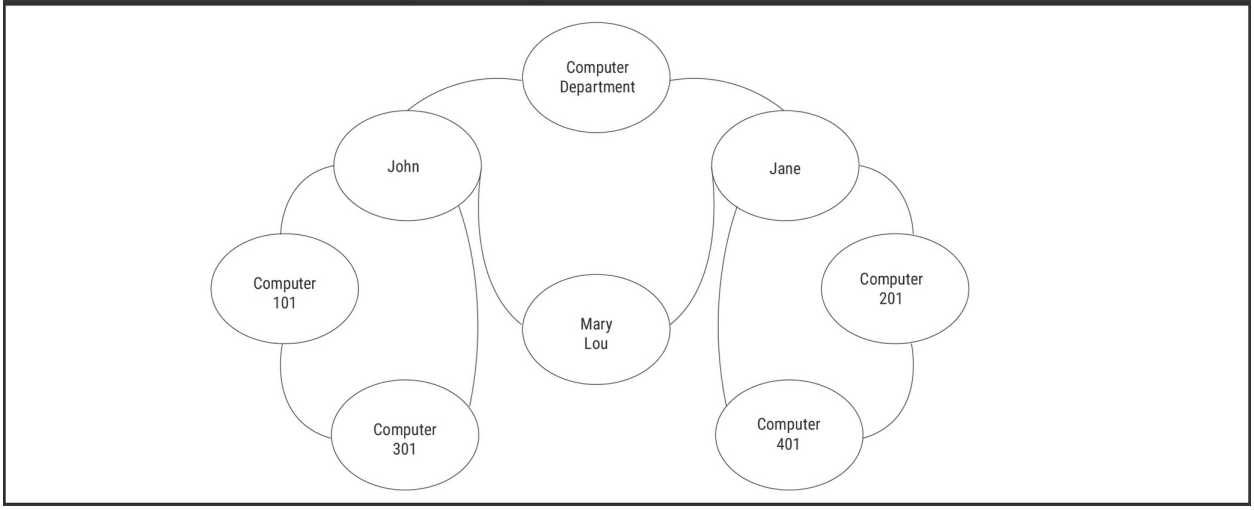
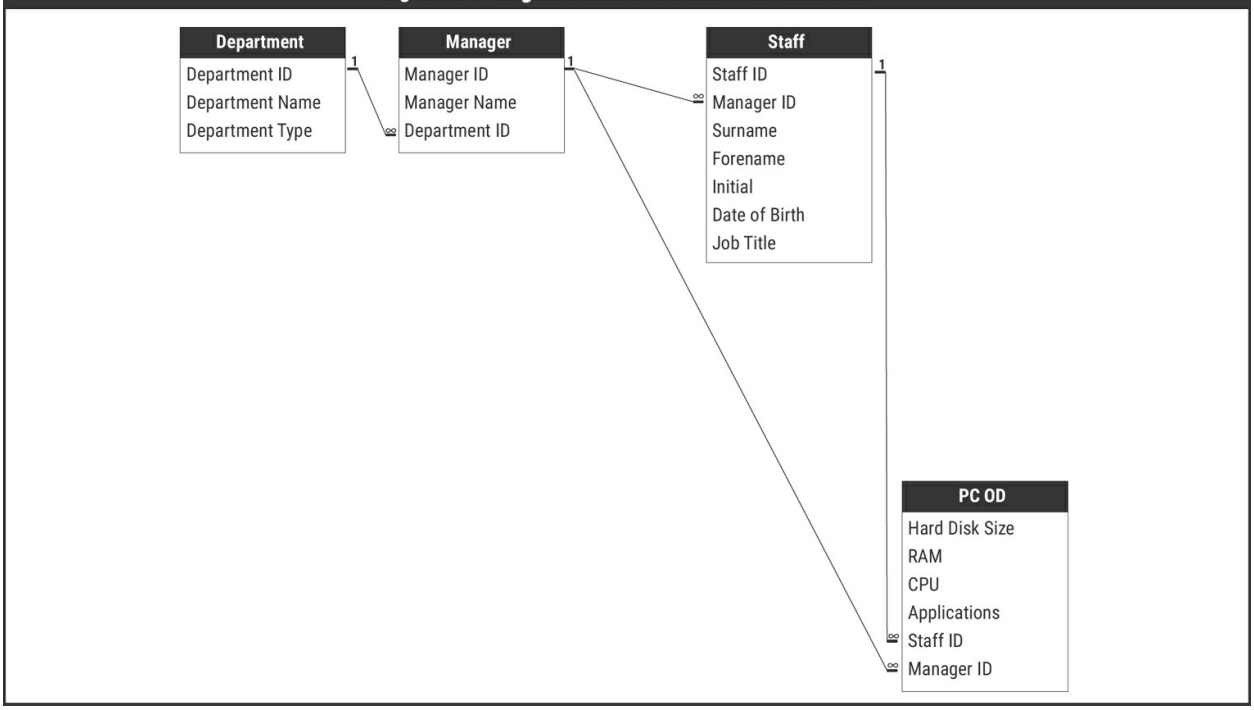


Figure 4.17—Organization of a Relational Database



- The sequence of columns is insignificant.
- The sequence of rows is insignificant.
- Each column has a unique name.

Certain fields may be designated as keys, so searches for specific values of that field will be quicker because of the use of indexing. If fields in two different tables take their values from the same set, a join operation can be performed to select related records in the two tables by matching values in those fields. This can be extended to joining multiple tables on multiple fields. These relationships are only specified at retrieval time, so relational databases are dynamic. The relational model is independent from the physical implementation of the data structure and has many advantages over the hierarchical and network database models. With relational databases, it is easier:

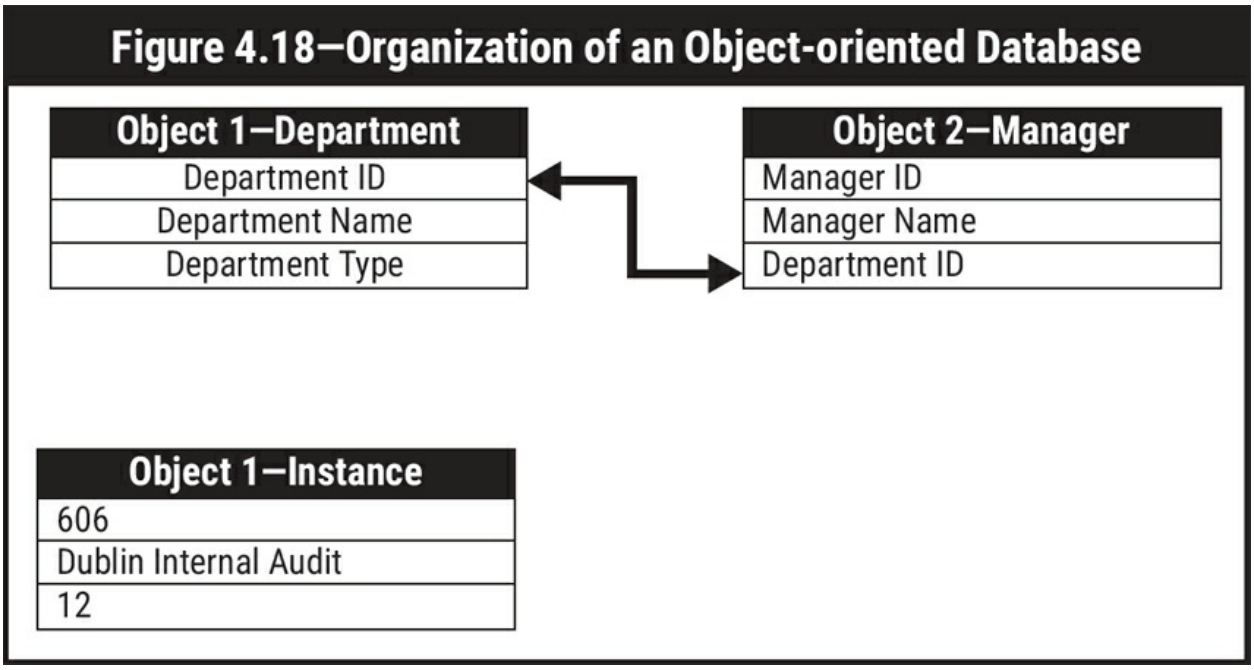
- For users to understand and implement a physical database system
- To convert from other database structures
- To implement projection and join operations (i.e., reference groups of related data elements not stored together)
- To create new relations for applications
- To implement access control over sensitive data

- To modify the database

A key feature of relational databases is the use of normalization rules to minimize the amount of information needed in tables to satisfy the users' structured and unstructured queries to the database. Generally followed, normalization rules include:

- A given instance of a data object has only one value for each attribute.
- Attributes represent elementary data items; they should contain no internal structure.
- Each tuple (record) consists of a primary key that identifies some entity, together with a set of zero or more mutually independent attribute values that describes the entity in some way (fully dependent on primary key).
- Any foreign key should have a null value or should have an existing value linking to other tables; this is known as referential integrity.

Object-oriented Database Management System (OODBMS)—An example of an OODBMS can be seen in **figure 4.18**. In an OODBMS, information is stored as objects (as used in object-oriented programming) rather than data (as in rational databases). This means that all of the features related to object-oriented programming can be applied, including encapsulation (i.e., the creation of data types or classes, including objects) and inheritance (i.e., classes inherit features from other classes). This results in objects that contain both executable code and data. The actual storage of the object in the database is achieved by assigning each object a unique identifier. These are loaded into virtual memory when referenced allowing them to be found quickly. OODBMS has found a niche in areas such as engineering, science and spatial databases, and is often used when the database is made up of graphics, diagrams or sound that cannot easily be defined or queried by relational databases.



NoSQL—NoSQL databases were developed in response to a rise in the volume of data stored on the Internet commonly known as big data. Much of these data are unstructured audio, video, tweets, logs, blogs, etc. These data cannot be broken out into components, which is required for a relational database; however, NoSQL databases may also support SQL, hence the term "Not only SQL." NoSQL databases may support object orientation (as per OODBMS) and other database technologies, shown in **figure 4.19**.

Figure 4.19—NoSQL Database Technologies

Data Model	Description
Key Value	All items in the database are stored as an attribute name (key) with its value.
Column-oriented	All of the values of a column are put together followed by all the values of the next column, then the values of the next column, etc.
Graph Database	Databases based on graph theory (mathematical models of the relationship between objects)
Document-oriented	Manages, stores and retrieves document-oriented information. This is achieved using storage methods, such as XML and JSON.

The advantages of NoSQL databases include sharding—the ability to

partition the database horizontally across database servers to spread the workload (important when dealing with big data)—and dynamic schemas—the schema does not have to be defined before you add data (as in relational databases). Common NoSQL databases include MongoDB and Cassandra.

4.11.3 DATABASE CONTROLS

It is critical that database integrity and availability are maintained. This is ensured through the following controls:

- Establish and enforce definition standards.
- Establish and implement data backup and recovery procedures to ensure database availability.
- Establish the necessary levels of access controls, including privileged access, for data items, tables and files to prevent inadvertent or unauthorized access.
- Establish controls to ensure that only authorized personnel can update the database.
- Establish controls to handle concurrent access problems, such as multiple users desiring to update the same data elements at the same time (i.e., transaction commit, locking of records/files).
- Establish controls to ensure accuracy, completeness and consistency of data elements and relationships in the database. It is important that these controls, if possible, be contained in the table/columns definitions. In this way, there is no possibility that these rules will be violated because of programming flaws or through the usage of utilities in manipulating data.
- Use database checkpoints at junctures in the job stream that minimize data loss and recovery efforts to restart processing after a system failure.
- Perform database reorganization to reduce unused disk space and verify defined data relationships.
- Follow database restructuring procedures when making logical, physical and procedural changes.
- Use database performance reporting tools to monitor and maintain database efficiency (e.g., available storage space, buffer size, CPU usage, disk storage configuration and deadlock conditions).
- Minimize the ability to use nonsystem tools or other utilities (i.e., those outside security control, to access the database).

4.11.4 DATABASE REVIEWS

When auditing a database, an IS auditor should review the design, access, administration, interfaces, portability and database supported IS controls, as shown in **figure 4.20**.

Figure 4.20—Database Reviews

Areas to Review	Questions to Consider
• Logical schema	• Do all entities in the entity-relation diagram exist as tables or views? • Are all relations represented through foreign keys? • Are constraints specified clearly? • Are nulls for foreign keys allowed only when they are in accordance with the cardinality expressed in the entity-relation model?
• Physical schema	• Has allocation of initial and extension space (storage) for tables, logs, indexes and temporary areas been executed based on the requirements? • Are indexes by primary key or keys of frequent access present? • If the database is not normalized, is justification accepted?
• Access time reports	• Are indexes used to minimize access time? • Have indexes been constructed correctly? • If open searches not based on indexes are used, are they justified?
• Database security controls	• Are security levels for all users and their roles identified within the database and access rights for all users and/or groups of users justified? • Do referential integrity rules exist and are they followed? • How is a trigger created and when does it fire? • Is there a system for setting passwords? Does change of passwords exist and is it followed? • How many users have been given system administrator privileges? Do these users require the privilege to execute their job function? • Has an auditing utility been enabled? Are audit trails being monitored? • Can database resources be accessed without using DBMS commands and SQL statements? • Is system administrator authority granted to job scheduler? • Are actual passwords embedded into database utility jobs and scripts? • Has encryption been enabled where required? • Are copies of production data authorized? • Are copies of production data altered or masked to protected sensitive data?

• Interfaces with other programs/software	• Are integrity and confidentiality of data not affected by data import and export procedures? • Have mechanisms and procedures been put in place to ensure the adequate handling of consistency and integrity during concurrent accesses?
• Backup and disaster recovery procedures and controls	• Do backup and disaster recovery procedures exist to ensure the reliability and availability of the database? • Are there technical controls to ensure high availability and/or fast recovery of the database?
• Database-supported IS controls	• Is access to shared data appropriate? • Are adequate change procedures utilized to ensure the integrity of the database management software? • Is data redundancy minimized by the database management system? Where redundant data exist, is appropriate cross-referencing maintained within the system's data dictionary or other documentation? • Is the integrity of the database management system's data dictionary maintained?
• IT asset management	• Has an owner been designated? • Have we retained a copy of the contracts/SLAs? • What is the license agreement? Are we in compliance with it?